A Comparative Approach between Batch and Online Machine Learning for Predicting Next-Minute Cryptocurrency Price Direction

Masters of Science Research Project
Data Analytics

Tiernan Barry
Student ID: x20199121

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Tiernan Barry
Student ID:	x20199121
Programme:	Data Analytics
Year:	2021
Module:	Masters of Science Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	16/08/2021
Project Title:	A Comparative Approach between Batch and Online Machine Learning for Predicting Next-Minute Cryptocurrency Price Direction
Word Count:	10,003
Page Count:	33

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Tiernan Barry
Date:	16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Comparative Approach between Batch and Online Machine Learning for Predicting Next-Minute Cryptocurrency Price Direction

Tiernan Barry
x20199121
Masters of Science Research Project in Data Analytics

16th August 2021

Abstract

As machine learning evolves within the era of big-data, this research addressed the topic of comparing batch machine learning against equivalent, online learning techniques in the context of next-minute cryptocurrency predictions. This research designed 3 comparative experiments, each predicting Ethereum, DogeCoin and BinanceCoin's next minute price direction: (Exp1.a) Batch Logistic Regression vs Online Softmax Regression, (Exp1.b) Batch Decision Tree vs Online Hoeffding Tree, and (Exp1.c) Batch Random Forest vs an Online Adaptive Random Forest. Exp1.a showed that across all 3 alt-coins, the Online Softmax Regression model outperformed the Batch Logistic Regression model, with the best online model accuracy and F1 being 0.61 compared to 0.56 for the Batch Logistic Regression. For Exp1.b, the results marginally favoured Batch Decision Trees over Hoeffding trees in 2 out of 3 datasets, with the best batch accuracy and F1 of 0.56 and 0.44, despite Hoeffding trees performing better on 1 alt-coin (BinanceCoin) with 0.63 accuracy. Exp1.c results favoured the Online Adaptive Random Forest in 2 out of 3 alt-coins, with the highest accuracy and F1 of 0.66 and 0.64 compared to 0.56 and 0.46, while Exp1.c results for DogeCoin marginally favoured the Batch Random Forest. Overall, model performances compare well with existing work on cryptocurrency predictions. Despite mixed results, this paper concluded that there are advantages of deploying online machine learning as opposed to batch learning for predicting next-minute price predictions of Ethereum, Dogecoin and BinanceCoin.

1 Introduction

This research contributes to the machine learning body of knowledge in the context of short-term (next-minute) cryptocurrency price predictions. Specifically, this investigation follows a comparative approach between traditional batch machine learning against online machine learning, with the goal of predicting next-minute price directions for DogeCoin (DOGE), BinanceCoin (BNB) and Ethereum (ETH) using historical pricing data and technical indicators (time series). As a result, this research also contributes to the broader body of literature regarding batch and online machine learning performance. A cryptocurrency is defined as a "digital currency" which facilitates financial transactions as a "means of exchange" (Mohapatra, Ahmed and Alencar 2019). Distinct from fiat currencies, cryptocurrencies are typically decentralised and are developed using open-source, distributed blockchain technologies for recording transactions (Islam et al. 2018). As of 2021, there are over 4,000 cryptocurrencies in circulation according to Investopedia (Conway 2021), while the original cryptocurrency – Bitcoin, released in 2009 – is considered the most successful, attracting the most investment, recognition and not surprisingly, research too (Sabry et al. 2020). Cryptocurrency prices are observed to be notoriously volatile compared to other tradable assets (Guo, A. Bifet and Antulov-Fantulin 2018); and much discussion continues around cryptocurrencies' underlying value, as well as to the role they truly play in today's economies' (Chiu and Koeppl 2017). Notwithstanding this, total cryptocurrency market capitalisation has sky-rocketed over recent years, particularly in the last 12 months seeing growth from 200 billion dollars in market capitalisation to over 2.3 trillion dollars according to Coinmarketcap.com (2021) (as of April 18th, 2021). As a result, cryptocurrency predictions pose as challenging, but highly interesting problems for data analytics research.

1.1 Research Question, Objectives and Contributions

From a machine learning perspective, batch learning is considered the "traditional" approach which describes how data arrives to train a model. As trends move towards "big-data", batch learning has been exposed as being limited for real-time analytics (Hoi et al. 2018). As a result, online machine learning is increasingly being utilised by researchers, but the relative comparison against batch learning has not been documented in the context of cryptocurrency predictions until this research. Therefore, by streaming cryptocurrency data from cloud storage (thereby simulating real-time analytics), this research project answers the following question:

Research Question: "To what extent can online machine learning provide an advantage over traditional batch machine learning techniques for researchers and investors when predicting next-minute cryptocurrency (alt-coin) price direction?"

In order to address this research question, the objectives listed in table 1 are completed. For objectives B-C, each machine learning model (ML) is implemented and evaluated:

Table 1: Research objectives

A. Process data	A.1. Create data lake of cryptocurrency data (1-min intervals) A.2. Identify which alt-coin prices to model using machine learning A.3. Process and transform data for predicting alt-coin prices
B. Batch ML	B.1. Batch Logistic Regression (B-LG-C) B.2. Batch Decision Tree Classifier (B-DT-C) B.3. Batch Random Forest Classifier (B-RF-C)
C. Online ML	C.1. Online Softmax Regression (O-LG-C) C.2. Online Hoeffding Tree Classifier (O-HT-C) C.3. Online Random Forest Classifier (O-RF-C)
D. Comparison	D.1. Individually compare batch and online algorithms D.2. Hypothesis test: Comparison of batch and online algorithms D.3. Compare results to literature
E. Dashboard	E.1. Develop cloud-based Shiny app to host results

This research contributes the first known clear-cut comparison between batch and online learning when applied to predicting next-minute cryptocurrency price movements. Due to the complex nature of cryptocurrency prices, this research is primarily limited by the scope of datasets examined (previous alt-coin pricing data). However, cryptocurrencies are likely influenced by a very complex combination of multiple factors not considered, such as social media, influential individuals, to name a few. Furthermore, this research is not proposing an actionable trading strategy as the focus is confined specifically to comparing learning algorithms across batch and online implementations.

For the remainder of this project, section 2 outlines the Literature Review of Machine Learning and Crypto-currency Analytics, section 3 covers the Methodology and Design, while section 4 showcases Implementation, Evaluation and Results. Section 5 covers the Discussion before concluding this research in section 6.

2 Literature Review of Machine Learning and Crypto-currency Analytics

This review is focused primarily on research conducted within the last 5 years. To the best of this work's knowledge, there is no literature which documents the comparison between online and batch learning in the context of cryptocurrency analytics. As such, this review is structured as follows: Introduction to Batch and Online Machine Learning (2.1); A Critical Review of Batch and Online Machine Learning (2.2) and A Critical Review of Cryptocurrency Analytics (2.3). Finally, Identified Research Gaps and Conclusion of Literature are presented in subsection 2.4.

2.1 Introduction to Batch and Online Machine Learning

Batch learning – also referred to as "offline learning" – is typically known as the "traditional" approach to machine learning and requires that all data is readily available to train a model at once (Hoi et al. 2018). Typically, the model is never updated with new training data as it is costly to implement, making batch learning unsuitable for the reality of many predictive problems, particularly in the presence of changing trends over time (concept drift) (Liu et al. 2016). Indeed, with the advent of big-data which is ever growing and changing, Hoi et al. (2018) notes that one of the "grand challenges" in the realm of Artificial Intelligence is scaling machine learning to be usable from continuous streams of data. This is consistent with Liu et al. (2016) who points out that batch learning fundamentally cannot cope with "large-scale datasets" due to memory constraints.

Accordingly, online learning – also referred to as incremental and/or data stream learning (Gomes et al. 2019) – can be deployed to overcome shortcomings associated with batch learning. Specifically, Rahnama (2014) outlines that online learning must exhibit the following characteristics: process and learn from data one record/instance at a time without reviewing the data again; consume limited computational resources and memory; sensitive to time constraints; and capable of generating a prediction at any given time. Hoi et al. (2018) notes that online learning provides advantages for many "real-world" scenarios, particularly where data is received "in a sequential order", such as financial markets time series. The key differentiator between these methods is that online learning continually predicts at each time step, while also learning from new data as it arrives.

2.2 A Critical Review of Batch and Online Machine Learning

Although studies have examined the comparative performance of batch and online learning across various domains, this has not been exhibited for cryptocurrencies. Addressing this comparative performance, Burlutskiy et al. (2016) used Stack Overflow online forums as a case study with the goal of predicting user response times to questions within the field of web-user behaviour. Accordingly, this comparison is achieved by running and evaluating online algorithms in mini-batches based on mean accuracy and time costs. As such, the researchers concluded that although a batch "deep learning" solution produced highest classification accuracy by small margins, it came with a training cost "several magnitudes higher" than basic online algorithms. Therefore, it was concluded that online models offer better prospects for real-time predictions, despite showing slightly compromised results, and that future research ought to trial "more datasets and prediction tasks" to test this performance comparison outside the scope of web-user behaviour, particularly where there is a need for real-time analytics, while mitigating the computational overhead.

Focusing on the Auto-regressive Integrated Moving Average (ARIMA) model, Liu et al. (2016) developed 2 novel online variants of ARIMA to overcome the problem of batch learning, aliased as ARIMA-ONS (ARIMA Online Newton Step) and ARIMA-OGD (ARIMA Online Gradient Descent). Due to the sequential nature of time series, the authors stipulate that online learning is a "more natural" approach for this task. Specifically, the authors report that during periods of "abrupt change" throughout the Dow Jones Industrial Average time series, ARIMA-ONS (online) reliably outperformed batch models, indicating a higher resilience to concept drift and volatility. This finding can be contrasted against Wang and Han (2014) who also predicted the DJIA index using batch and online models. In particular, a specific neural network architecture – Extreme Learning Machine (ELM) (Liang et al. 2006) – was tested against its online sequential variant (aliased as OS-ELMK), as well as an online Support Vector Regressor (OS-SVR). The results show that OS-ELMK has multiple advantages: it continuously adapts to new samples; OS-ELMK can predict with very similar performance as the batch ELM, but with orders of time complexity reduced. Distinct from Liu et al. (2016), the online accuracy for their stock forecasting did not surpass the batch model. Such conflicting results indicate that more work is necessary, using more algorithms/datasets.

In the healthcare domain, Jagirdar (2018) investigated online and batch learning performances (binary classification) across the criteria of "accuracy, model complexity and time consumption". By examining two datasets: patient mortality (9,000+ observations) and diabetic burnout (100,000+ observations). Although a series of batch and online models were trialed showing mixed results across datasets, only the Random Forest model was implemented in batch and online learning. Using the Area Under the Curve (AUC) metric, the relevant results from this study showed that batch random forest classifier achieved 0.76 AUC, while using online random forest returned an impressive 0.98 AUC using the mortality dataset. By contrast, the diabetic burnout dataset inquiry did not vary between batch and online having a 0.98 AUC across both implementations. Similar to Burlutskiy et al. (2016), Jagirdar (2018) noted that future work is needed using more online algorithms for real-time analytics.

The topic of batch and online learning is framed by Suto et al. (2017) as a mix of Internet of Things (IoT) with machine learning. In the context of human activity recognition (HAR) – where subjects wear sensors for monitoring human activity – this research set out to test the "efficiency and reliability" of successfully implemented batch machine learning techniques when applied under real-time conditions. By comparing their real-time results to previously recorded results across literature, the authors note that their online learning tests did not reach the levels observed by other researchers using batch/offline methods. Finally, from a climate research perspective, Brenowitz et al. (2020) deployed machine learning techniques to predict temperature and humidity. Although part of a broader effort to improve the accuracy of traditional approaches of climate forecasting, this research implemented a Random Forest and 2 layered Neural Networks in both batch and online learning modes. This research finds that the neural network outperforms the random forest model in offline mode. However, the neural network is described as being too unstable for online modelling, thereby giving credit to the random forest model for its real-time stability. However, more research is needed to find offline predictive modelling techniques which "translates to good online performance" (Brenowitz et al. 2020).

2.3 A Critical Review and Comparison of Cryptocurrency Analytics

The novel nature of cryptocurrencies has attracted research from multiple angles, covering a wide range of datasets, algorithms and methodologies; yet the majority of contributions have studied Bitcoin price predictions (Sabry et al. 2020). According to the objectives set out in 1.1, a series of papers are reviewed which utilise machine learning techniques for predicting cryptocurrencies in general, which is justified as most research is Bitcoin-centric. To begin, J.-Z. Huang, W. Huang and Ni (2019) raised an important point which distinguishes cryptocurrency from stock predictions: cryptocurrencies are not linked to fundamental factors (ie. profits, balance sheets, etc), therefore much of the research reviewed utilise technical factors (previous cryptocurrency prices) to infer future prices/direction. In fact, technical factors have been shown to be the most promising predictor variables for cryptocurrencies (Jaquart, Dann and Weinhardt 2021).

Focusing on real-time predictions, Amjad and Shah (2017)'s research into Bitcoin price predictions was seeking to forecast price changes for algorithmic trading. Accordingly, Amjad and Shah (2017) achieved 70+ percent classification accuracy, as well as significant profit returns by deploying Logistic Regression, Random Forest and Linear Discriminant Analysis using 2014-2016 data. By contrast, also studying Bitcoin price predictions, McNally, Roche and Caton (2018) developed ARIMA, Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) models for both classification and regression tasks between 2013-2016 using daily prices. As anticipated, LSTM and RNN outperformed the ARIMA model for both classification and regression, despite 52 percent accuracy being the highest achieved result (LSTM model). The authors note that limitations and future work areas include using data streaming which they expect to improve model performance. In conjunction with this, McNally, Roche and Caton (2018) highlight that higher frequency data (ideally, minute-minute level) would be likely to improve on deep learning models such as LSTM and RNN implementations. Using higher frequency datasets to improve model performance is a commonly held claim for predicting cryptocurrencies (Sebastião and Godinho 2021; Munim, Shakil and Alon 2019; Pang, Sundararaj and Ren 2019). Indeed, research has been done using higher-frequency datasets (highest being 15 minute intervals) studying the classification of 12 alt-coin price returns, with the goal of testing and "exploiting" market inefficiencies Akyildirim, Göncü and Sensoy (2018). The best performing algorithms (using out-of-sample results, and 15-minute intervals) appear to have been 0.64 accuracy using an MLP Artificial Neural Network, predicting the OMG-Network alt-coin returns, while all tests return at least 0.5 accuracy.

Framing this task into a multinomial classification problem (21 classes, categorised into percentage increase/decreases), J.-Z. Huang, W. Huang and Ni (2019) deployed a decision tree model on a "high dimensional" dataset of 124 variables containing daily records from 2012-2017. This research portrayed the utility of previous price indicators (technical analysis) for predicting Bitcoin, as the results were able to outperform the "buy-and-hold strategy even in a strong bull market". However, a number of research papers have emphasised that there is a lack of contributions which utilise streaming data and/or real-time predictions with the cryptocurrency space (Mohapatra, Ahmed and Alencar 2019; Jay et al. 2020; Horvat et al. 2020). This gap in literature led Mohapatra, Ahmed and Alencar (2019) to develop a scalable Bitcoin prediction implementation using high frequency cryptocurrency data (1-minute intervals), while also using Twitter data from July-August 2019. The authors aliased their solution as 'KryptoOracle'. While KryptoOracle mainly focused on text analytics at first, the authors later incorporated historical Bitcoin prices to increase model performance, finding that model performance improves over time showing the adaptive nature in real-time. Mohapatra, Ahmed and Alencar (2019) identified future work of developing more machine learning techniques, particularly using streaming algorithms which were not available to them (ie. non-linear models).

In a comprehensive review of 100 alt-coins, Liew et al. (2019) modelled the daily returns using 11 algorithms (linear, tree, ensemble, deep learning models). Among the findings, the authors demonstrated that smaller, more volatile cryptocurrencies were the hardest to predict, while larger, less volatile performed better. Analysing the Ethereum alt-coin, Chen, Narwal and Schultz (2019) also deployed multiple models to the task of classifying price direction using hourly data, which the authors proclaim is "non-trivial" and challenging task for machine learning. Across 10 deployments, the best performer was an ARIMA model with 0.61 accuracy, while all other models ranged between 0.5-0.57 accuracy, Random Forest being the worst. Seemingly the best performing model identified in this review, Albariqi and Winarko (2020) implemented a LSTM for predicting Bitcoin price directions. The dataset ranged from 2010-2017 with records for every second day (1300 rows). Albariqi and Winarko (2020) reported mean accuracy of 81.3 percent, precision of 81 percent and recall as high as 94 percent. However, this research is limited due to small dataset size, and the fact that predictions are for 2-60 days, rather than providing any real-time, actionable insight. This critique is consistent with Jaquart, Dann and Weinhardt (2021) who claims that not enough researchers focus on short-term predictions, and that there is much to be discovered regarding Bitcoin price predictions within short time horizons (under 60-minutes). It is fair to assume this extends to alt-coin prices too. Therefore, Jaquart, Dann and Weinhardt (2021) predicted 1, 5, 15 and 60 minute price movements with multiple algorithm types, finding that 60-minute predictions performed best returning 0.56 accuracy (LSTM). For 1-minute predictions, the best is also

LSTM with 0.52 accuracy.

Claiming that current literature doesn't address the problem of real-time predictions, Jay et al. (2020) studied Bitcoin, Litecoin and Ethereum from 2017-2019 (daily prices) from the perspective of volatility due to the "erratic" behaviour of these markets. Using a selection of neural networks, Jay et al. (2020) compared stochastic induced models with "deterministic" models. This is based on the premise that real-time predictions require the element of random/stochastic behaviour. Jay et al. (2020) reported that model accuracy is improved by 1.3 to 3.9 percent utilising the stochastic induced networks. Incremental learning using a batch size of 7 was implemented to train the networks.

2.4 Identified Research Gaps and Conclusion of Literature

Combining both bodies of literature, the following gaps and limitations are identified: (a) First, a common theme pointed out by most of the papers in 2.2 confirm that there is a need and benefit of continued research into the comparison between batch and online models using more datasets, algorithms and across different domains (Burlutskiy et al. 2016; Jagirdar 2018; Brenowitz et al. 2020). This is also supported by Hoi et al. (2018) who outlined the "grand challenge" of bringing model usability to "continuous data streams". (b) Taking the first research gap into account, there are currently no known studies which examine cryptocurrency price predictions using batch and online learning techniques, despite this being done across other research domains discussed in 2.2. Furthermore, within the cryptocurrency literature discussed, authors have identified that future work should consider streaming data and online learning algorithms to improve results, however no controlled comparison has been done against the traditional, batch models to warrant the use of online and data streaming models (Mohapatra, Ahmed and Alencar 2019; McNally, Roche and Caton 2018). (c) Next, the majority of cryptocurrency research papers focus on Bitcoin priced in US Dollars (Sabry et al. 2020). Accordingly, this research focuses on prediction of 3 alt-coins instead using Euro currency. (d) The fourth issue identified is regarding the dataset sizes being examined, which typically do not extend greater than daily price records (Jaquart, Dann and Weinhardt 2021; McNally, Roche and Caton 2018; Sebastião and Godinho 2021; Munim, Shakil and Alon 2019; Pang, Sundararaj and Ren 2019). This is potentially highly problematic for many machine learning techniques as the models may need more data to learn intra-day patterns in the data. This is also important from a stakeholder perspective, as daily price predictions do not lend themselves to actionable decisions (ie. when in the day should one buy/sell?). All considered, this research develops the first known comprehensive account of using streaming and online techniques versus the traditional batch approach for alt-coin predictions.

3 Methodology and Design

3.1 Batch versus Online Machine Learning Methodology

This section outlines the steps taken throughout this research, while section 4 details the implementation with evaluation and results. When embarking on data analytics research, one is confronted with a series of possible methodology frameworks to consider. Most notably, these include the Cross Industry Standard Process for Data Mining (CRISP-DM) and the Knowledge Discovery in Databases (KDD) paradigms (Islam et al. 2018). Given this research is not within a business setting, a tailored version of the Knowledge Discovery in Databases (KDD) is best suited to this task, which is hereby named "Batch versus Online Learning Methodology" as visualised in figure 1, and also within the "Analytics Tier" of figure 7 highlighting the interplay between methodology and design. Accordingly, the methodology consists of the following stages: (1) Select Cryptocurrency Data; (2) Pre-process Cryptocurrency Data; (3) Transform Cryptocurrency Data; (4) Batch versus Online Machine Learning; (5) Knowledge Discovery.

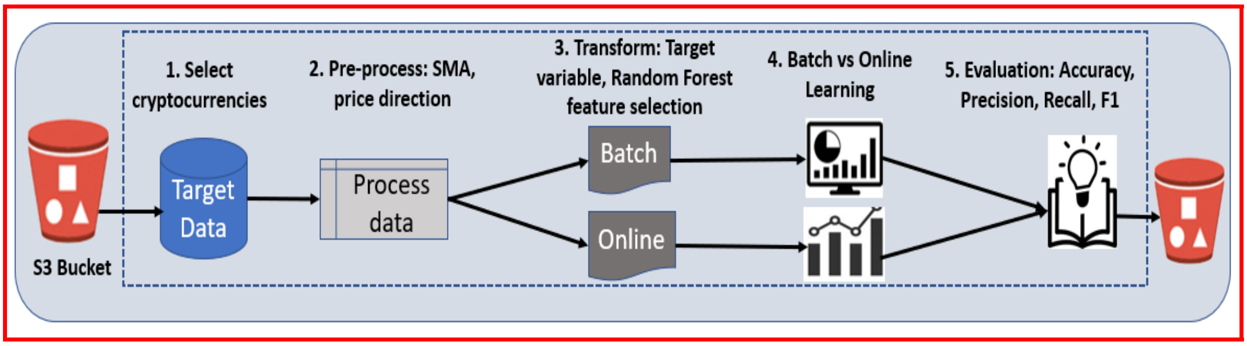


Figure 1: Batch versus Online Learning Methodology (Analytical Tier)

3.1.1 Select Cryptocurrency Data

First, this research extracted[1] all cryptocurrency prices from the Binance exchange priced in Euro currency (table 2, 29 cryptocurrencies, 1-minute intervals). The following fields are extracted: Timestamp, Open, High, Low, Close, Volume, Quote Asset Volume, Trades, Taker Buy Base Asset Volume, Taker Buy Quote Asset Volume. By extracting a comprehensive set of cryptocurrencies, this enabled a well informed process for deciding to model DogeCoin, BinanceCoin and Ethereum price movement.

Table 2: Cryptocurrency datasets

ADA/EUR	AVAX/EUR	BCH/EUR	BNB/EUR	BTC/EUR
BTT/EUR	CHZ/EUR	DOGE/EUR	DOT/EUR	EGLD/EUR
ENJ/EUR	EOS/EUR	ETH/EUR	GRT/EUR	HOT/EUR
LINK/EUR	LTC/EUR	LUNA/EUR	MATIC/EUR	SXP/EUR
THETA/EUR	TRX/EUR	UNI/EUR	VET/EUR	WIN/EUR
WRX/EUR	XLM/EUR	XRP/EUR	YFI/EUR	

The following criteria is applied for selecting DogeCoin (DOGE/EUR), BinanceCoin (BNB/EUR) and Ethereum (ETH/EUR) to study: (a) Bitcoin is excluded as it is widely researched to date; (b) Cryptocurrencies with less than the median dataset size (circa 210,000 records) were automatically dropped (14 cryptocurrencies), due to the preference of using larger datasets where possible (Sebastião and Godinho 2021; Munim, Shakil and Alon 2019; Pang, Sundararaj and Ren 2019). Accordingly, there is over 222,000 rows of DogeCoin minute prices, while BinanceCoin and Ethereum have over 780,000 rows; (c) Next, this research sought to study a high, medium and low volatility time series calculated by the standard deviation of the price percentage change (figure 2). This identified that Ethereum is the least volatile alt-coin (0.0017), while DogeCoin is the most volatile

[1]Credit to Nistrup (2019) for providing online article using the Binance API which helped this methodology

(0.00371). BinanceCoin is included as showing a moderate volatility level (0.00276). By selecting these 3 large cryptocurrency datasets, this addresses research gaps (a) and (b) while also ensuring a range of complexity. Objectives A.1 and A.2 are now completed.

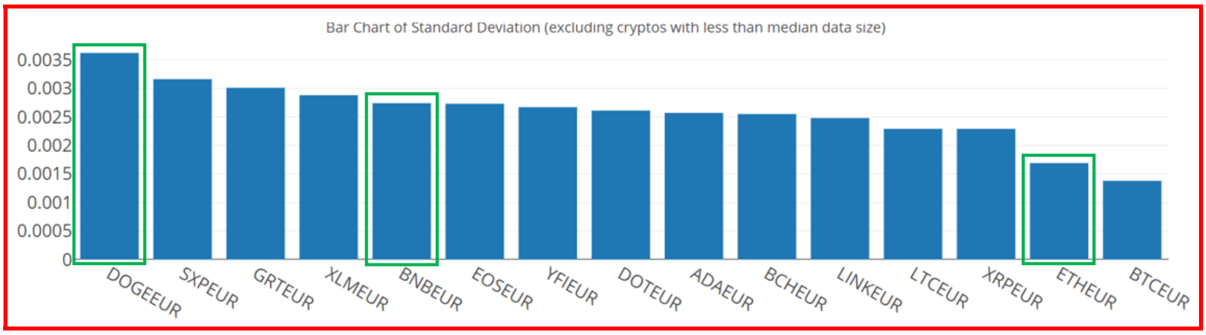


Figure 2: Standard Deviations of Price Changes (excluding 14 cryptocurrencies)

3.1.2 Pre-process Cryptocurrency Data

As outlined in figure 1, data pre-processing is identical for both batch and online learning as part of exploratory analysis (Ethereum, DogeCoin, BinanceCoin). Simple moving averages (SMA) are generated for 10, 20, 50, 100, 200, 300 minute intervals. SMAs are widely used technical indicators which indicate trends in the data, and can be used to predict future prices J.-Z. Huang, W. Huang and Ni (2019). Because this is done at the minute-minute level, it is hard to discern the various time series from one another without zooming in on a 12-hour window as shown between figures 3 and 4.

Figure 2: DogeCoin (DOGEEUR) Time Series

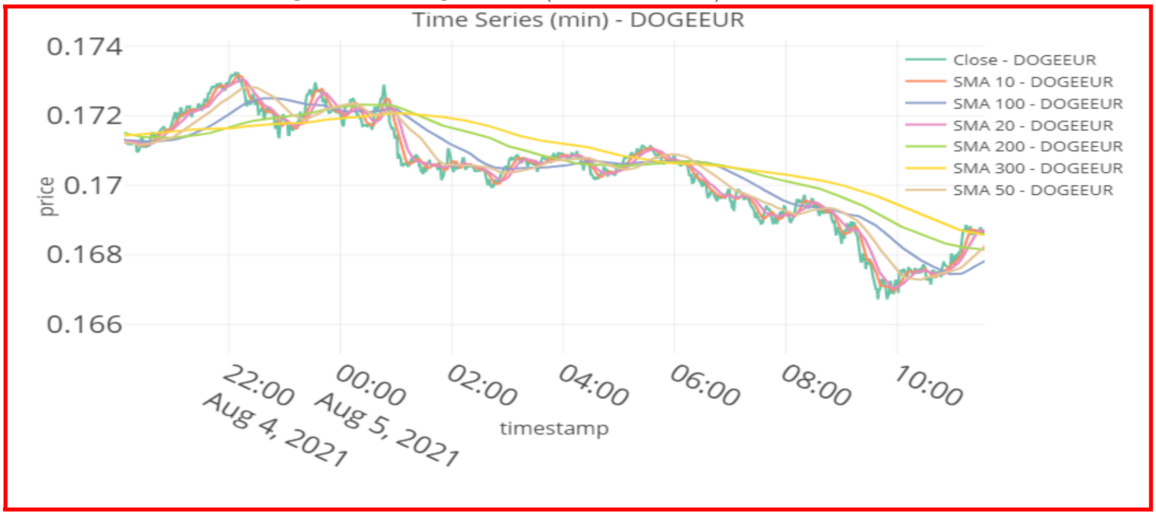


Figure 3: DogeCoin (DOGEEUR) Zoomed Time Series

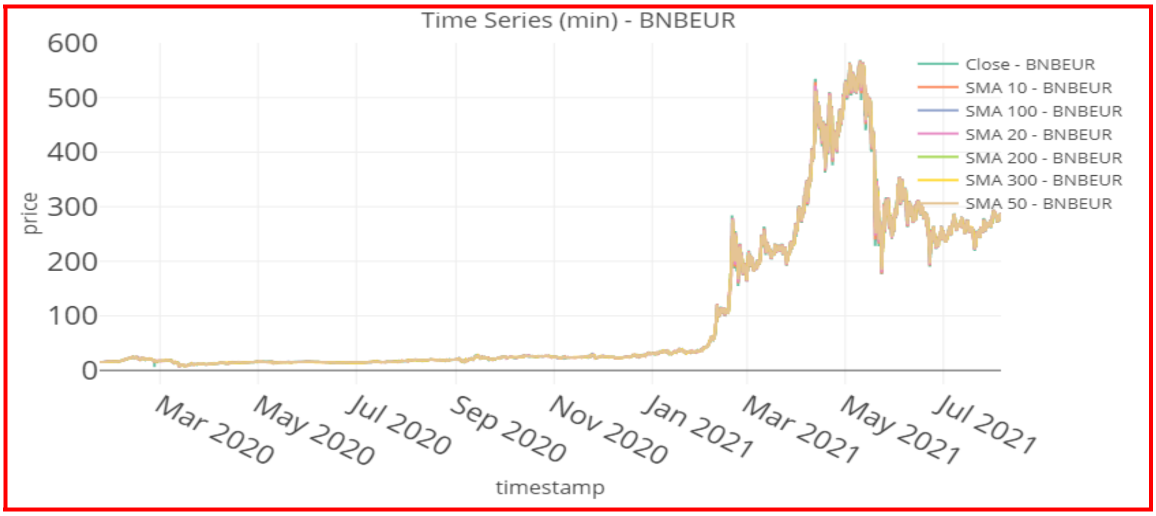


Figure 4: BinanceCoin (BNBEUR) Time Series

Price direction is generated for every minute, which enables this analysis to implement classification tasks from the time series data. The price direction field is populated with '1' where the price increases compared to the previous price, '0' when prices remains the same, while -1 is labelled for a decrease in price. Subsequently, class imbalance is identified for each cryptocurrency (figure 5).

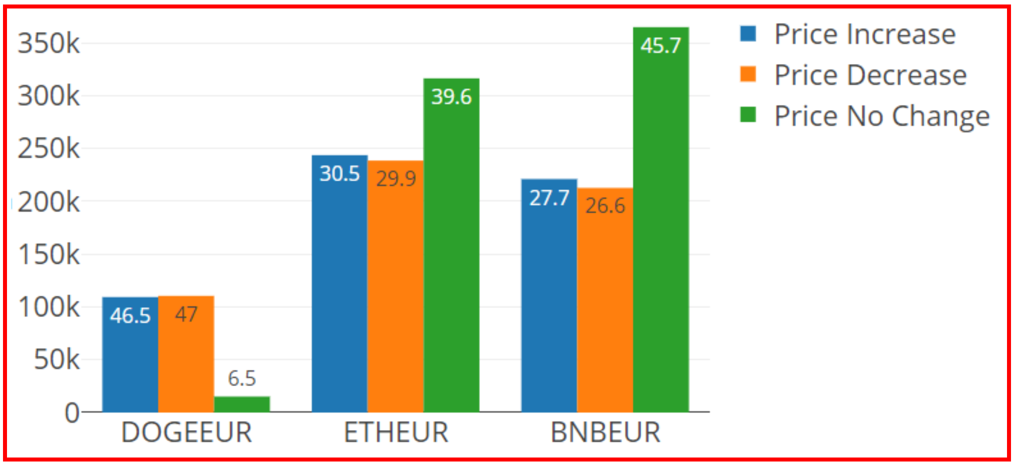


Figure 5: Class Balance for DogeCoin, Ethereum, BinanceCoin

3.1.3 Transform Cryptocurrency Data

By addressing the pre-processing findings, this subsection completes objective A.3 of preparing the data for machine learning with the following activities:

Create target variables: The target variable to be predicted across all experiments is the next minute price direction. Therefore, once the price direction is generated as described in 3.1.2, this is shifted back by 1 row when training machine learning models. This is because it is required to predict the future price direction, given the current information. For example, at time 't', a model needs to predict the price for time 't+1'.

Feature selection: With the goal of classifying price direction with input data which is fundamentally continuous (ie. continuous input, categorical output), many commonly used feature selection techniques such as correlation analysis are invalid. Therefore, this approach uses machine learning to help identify and/or rule out features. Accordingly, the Random Forest feature importance technique is used, which plots the following features importance's for each alt-coin, as shown in figure 6 for Ethereum:

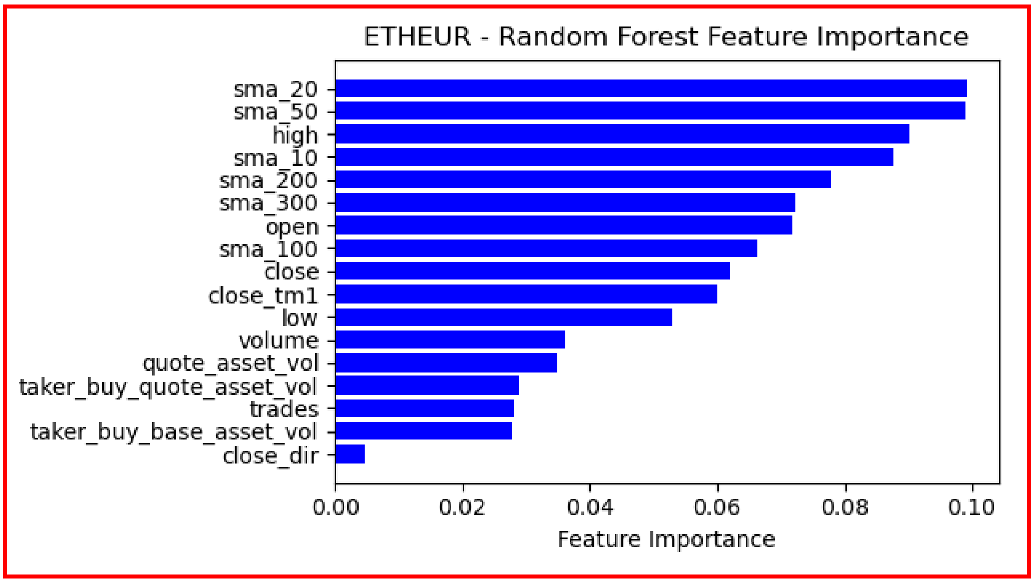


Figure 6: Feature Selection

When applied to each alt-coin, it is clear that there is no 1 or 2 features which dominate the learning process (as shown in below table, the highest contribution is 0.12), therefore it is decided to exclude only the features in the lowest quartile. Following this approach, 87 percent of feature importance is captured for Etheruem and BinanceCoin, while DogeCoin's is 78 percent. Table 3 summarises the features selected for deployment per alt-coin:

Table 3: Ethereum, DogeCoin and BinanceCoin Feature Importance's

	FEATURE	IMP		FEATURE	IMP		FEATURE	IMP
12	sma_10	0.102	11	volume	0.071	12	sma_200	0.122
11	sma_20	0.101	12	quote_asset_vol	0.071	11	sma_300	0.121
10	sma_50	0.096	10	sma_300	0.07	10	sma_100	0.111
9	high	0.086	9	sma_200	0.069	9	sma_50	0.088
8	close	0.085	7	taker_buy_base_asset_vol	0.068	8	high	0.083
7	sma_100	0.079	8	taker_buy_quote_asset_vol	0.068	7	sma_20	0.077
6	sma_300	0.076	6	sma_100	0.066	6	sma_10	0.063
5	sma_200	0.069	5	sma_50	0.065	5	quote_asset_vol	0.046
4	open	0.058	4	sma_20	0.062	3	close	0.042
3	close_tm1	0.053	2	sma_10	0.06	4	open	0.042
2	quote_asset_vol	0.037	3	trades	0.06	2	volume	0.039
1	volume	0.035	1	close	0.053	1	close_tm1	0.038

Class imbalance: Given the time series nature of this research (that is, each sample is not independent of the previous), it is not palatable to transform the data using over/under sampling techniques, as this will inherently violate the time component of the data, and create a "non-representative" dataset for training (Cao et al. 2013). Instead, each batch trained model is provided with the class weight parameter, which is calculated by weighting the classes while training the model with their inverse proportion to the current class distribution. Because the class distribution cannot be known in advance for online tasks, online learning models use the Hard Sampling Classifier technique (from the 'river' package), which stores a fixed size of samples which are deemed the most difficult to classify by the algorithm, and uses this data to train the model at a specified probability (Montiel et al. 2020a).

3.1.4 Batch versus Online Machine Learning

Evidently, this research empashised the use of tree-based machine learning models (though, not exclusively), which is justified given that these model types are among the least utilised in current cryptocurrency literature (Sabry et al. 2020). While each implementation is covered in section 4, a cursory overview of each model is provided:

Logistic and Softmax Regression Logistic regression is the most commonly used approach for predicting categorical variables (particularly for binary classes) (Burkov 2019; Hilbe 2011). The concept underpinning logistic regression revolves around the "odds" of an event occurring or not, which logistic regression models as a "linear function" of predictor variables using the natural logarithm of the odds (log-odds) to predict the target class (Burkov 2019; LaValley 2008). Overall, logistic regression is noted as being an efficient and scalable model for classification tasks (Amjad and Shah 2017). Softmax regression is defined as a multinomial "generalisation" of the logistic regression model, and therefore can be used for multi-class problems (Jurafsky 2020; Montiel et al. 2020b).

Decision Tree A decision tree is a highly intuitive and "white-box" model (as opposed to black-box, meaning it is easy to understand) which learns rules from the data in order to predict the target class/variable and is used in both classification and regression problems (Pedregosa et al. 2011a). There are many variations of the decision tree algorithm (eg. ID3, C4.5, CART), however each of these require that all data is readily available to train from memory, which therefore inhibits their utility for big datasets, and rules them out for data stream analytics altogether (Domingos and Hulten 2000).

Hoeffding Tree The Hoeffding tree algorithm is also a tree-based model and was developed to address the limitations of traditional decision trees (Domingos and Hulten 2000). Accordingly, the Hoeffding tree is in effect a decision tree which can be deployed using online learning and can be used for real-time analytics (Domingos and Hulten 2000). It does so by constructing a decision tree using Hoeffding bounds enabling the algorithm to converge to the same level of a batch trained decision tree provided enough streams of data (Domingos and Hulten 2000).

Random Forest The random forest algorithm is a widely used example of an "ensemble" machine learning algorithm (Burkov 2019). Ensemble learning is a technique which seeks to build a highly accurate model by combining several "weak" or "shallow" learning models (models that are not deep learning neural networks) into an aggregated model. Effective for both classification and regression tasks, random forests are constructed of multiple decision tree algorithms, and are more resilient to over-fitting compared to decision trees due to the bagging technique which ensures each decision tree learns on slightly different copies of the original data. Random forests predict classifications by returning the class which the majority of decision trees predicted, and are therefore considered highly robust models (Burkov 2019; Amjad and Shah 2017).

3.1.5 Knowledge Discovery

The following evaluation metrics are calculated using the weighted average approach rather than using the default 'macro', or 'micro' function. This is justified due to class imbalances in the data, meaning that each of the following metrics are calculated for each class (1, 0, -1), and then weighted by the support level of the class. For batch learning models, the average metric for each cross validation split is used in results. Because online learning metrics are continually updated, the final (ie. the current) value of each metric is used. All of the following are interpreted between 0 (worst) and 1 (best).

Mean Accuracy Mean accuracy captures the ratio of correct predictions compared to all predictions made (Hossin and Sulaiman 2015).

Precision Out of all the samples that the classifier predicted as being positive, precision measures how many of them were correctly identified (Hossin and Sulaiman 2015).

Recall Out of all the positive samples that actually exist, recall measures how many of them were correctly identified (Hossin and Sulaiman 2015).

F1 Score The F1 score returns the "harmonic mean" of the precision and recall metrics, and is often compared to mean accuracy but is considered preferable under conditions where false predictions are more crucial (Hossin and Sulaiman 2015).

3.2 3-Tier Batch versus Online Machine Learning Design

This research developed a 3-tier architecture (as shown in figure 7) owing to fact that a significant level of development was necessary in order to create a live, voluminous dataset while also using scalable, cloud technologies in a cost effective manner. The interplay between methodology and design is summarised in figure 7:

Data Persistent Tier: The purpose of the Data Persistent Tier is to continually generate live cryptocurrency data into an AWS Simple Cloud Storage bucket (S3), where the data can be loaded/streamed into the Analytics Tier. Everyday, data is collected and loaded into AWS S3 buckets using the Binance's API (binance.client) programmed through Python 3.8.10, which is scheduled to run using Ubuntu's cron utility, hosted on AWS EC2 cloud instances. The following EC2 instance is utilised:

- EC2 Region/Availability Zone: US East, N. Virginia (us-east-1b).
- EC2 AMI: Ubuntu Server 20.04 LTS (HVM), SSD Volume Type.
- EC2 Instance type: t.2 large (2 VCPUs, 8GB RAM).

- Storage: 16GB of Elastic Block Storage (EBS).
- Security: Ports 20 and 80 are open for custom IP addresses only (HTTP and SSH).

In turn, this EC2 instance is scheduled to run using the AWS Instance Scheduling service comprised of AWS CloudFormation, CloudWatch, Lambda and DynamoDB.

Analytics Tier: The Analytics tier is the development environment where pre-processing and analytical techniques are deployed to address the research question. The primary programming language used is Python 3.7.10 running from Pycharm Community Edition 2019.2.3, using Windows 10 with 16GB of RAM and an Intel Core i5-7300HQ CPU.

Visual Tier: The Visual Tier is developed to inspect analysis and results in an interactive manner, hosted on a second EC2 instance. The programming language utilised for this web-application is R version 3.6.3, using the shiny package (version 1.6.0). Similarly, this EC2 instance specification is the same as the Data Persistent Tier, except for the following differences:

- Storage: 8GB of Elastic Block Storage (EBS).
- Security: Ports 20, 80, 8787 and 3838 are open for custom IP addresses only. These are used for SSH, HTTP, R-Studio and the Shiny app service respectively.

Because each tier is running from completely different host systems, this means that code has to be managed in an effective manner. To do so, the Git version control utility is used to track the code on each system and Github is used as the remote repository.

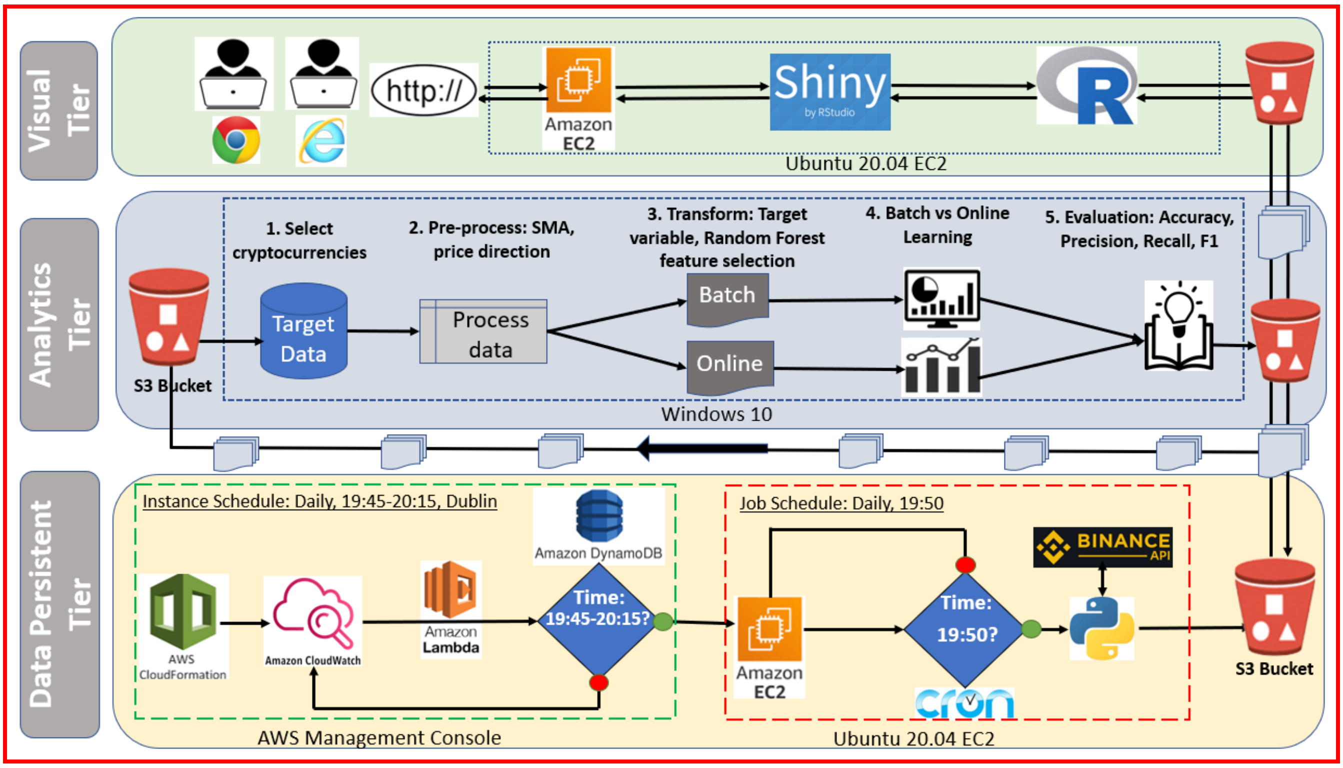


Figure 7: Batch versus Online Learning 3-Tier Analytical Architecture

4 Implementation, Evaluation and Results of Batch versus Online Learning

This section documents the experimental implementation to address the research question posed in subsection 1.1, consisting of batch and online machine learning across each alt-coin. Accordingly, the following experiments were completed:

Experiment 1: The main component of this implementation was to individually compare the predictive performance (using the evaluation metrics in subsection 3.1.5) of each model when deployed under batch and online conditions. By completing the following experiments, this addresses objective D.1: (Exp.1a) Batch Logistic Regression vs Online Softmax Regression; (Exp.1b) Batch Decision Tree vs Online Hoeffding Tree; (Exp.1c) Batch Random Forest vs Online Random Forest.

Experiment 2: Unpaired Two-Sample Hypothesis Test: Using model evaluation metrics collected from Experiment 1, a hypothesis test was performed for determining whether a differential between batch or online learning predictive performance is statistically significant. By completing this secondary experiment, this addresses objective D.2.

4.1 Experiment 1 Implementation, Evaluation and Results

Before progressing to Exp.1a-c, a brief overview of Experiment 1 is presented here. While this implementation delivers a comparison between batch and online models, all model development is conducted using batch data. Once the final batch model is identified in Exp.1a-c (using grid-search and time series cross validation), a replica of this model (ie. hyper-parameters, features, etc) is deployed using online learning where applicable. This is justified from a machine learning operations (ML-ops) perspective, and also provides scientific control from which to compare results. For online learning deployments, data is streamed from an AWS S3 CSV file, transformed into a python dictionary object, and then fed to models for online learning. Figure 8 summarises the implementation of Experiment 1 (Exp.1a-Exp.1c) for each alt-coin and model.

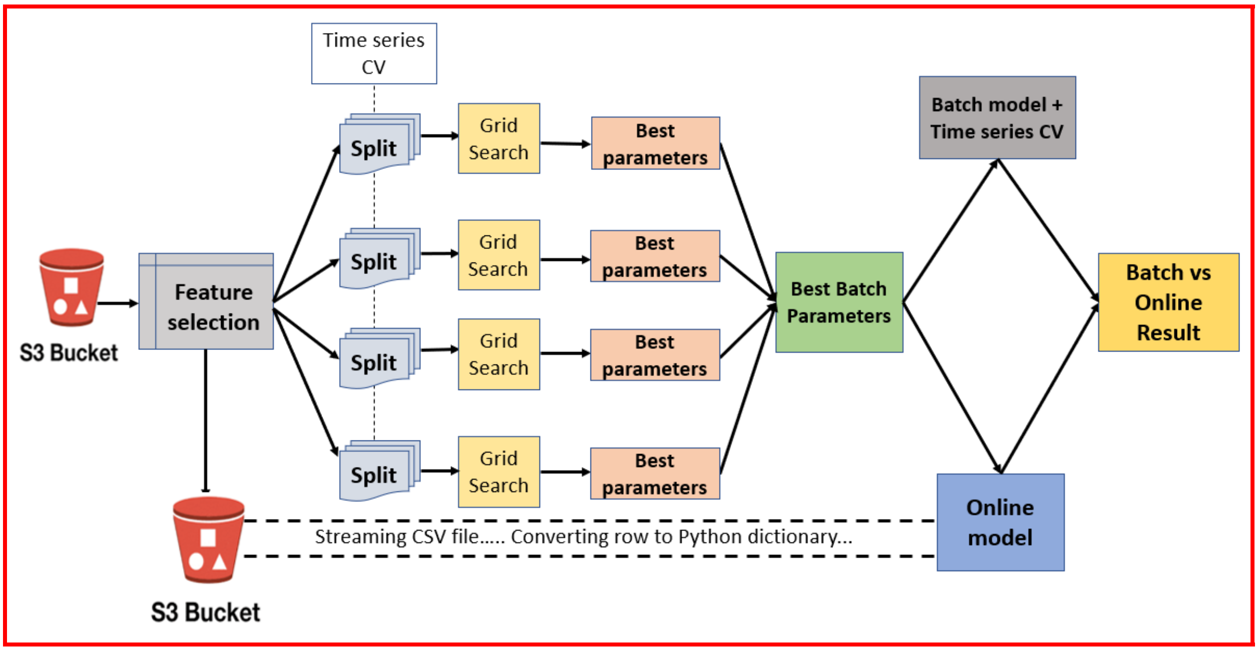
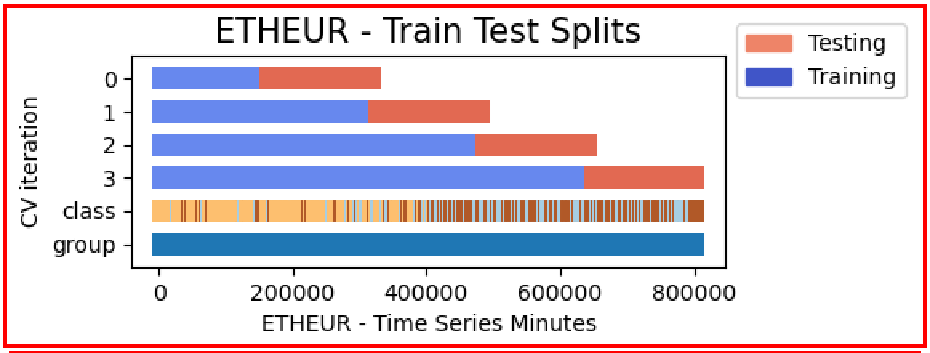
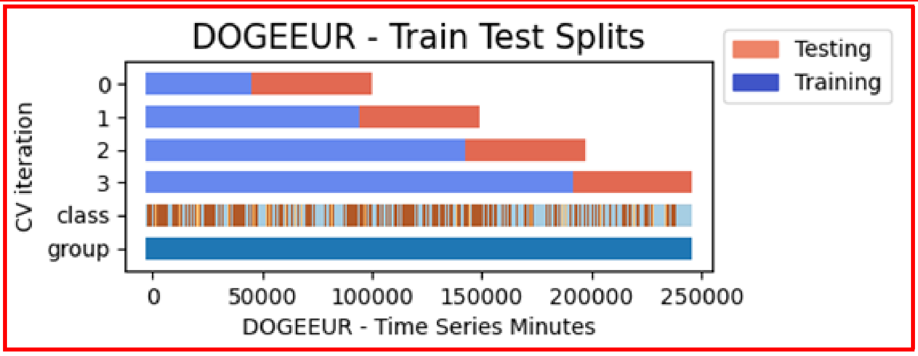


Figure 8: Experiment 1 Flow Chart

Secondly, the time series cross validation splits, including class balance, is plotted for each alt-coin throughout figure 9 (plotting technique adopted from the scikit-learn documentation/code (Pedregosa et al. 2011b)). This approach maintains the sequential nature of the time series, and ensures the model is never trained on future data to predict past prices. The 'class' bars also reveal the sequentially imbalanced and dynamic nature of the data as time passes (particularly for BinanceCoin and to a lesser degree, Ethereum).





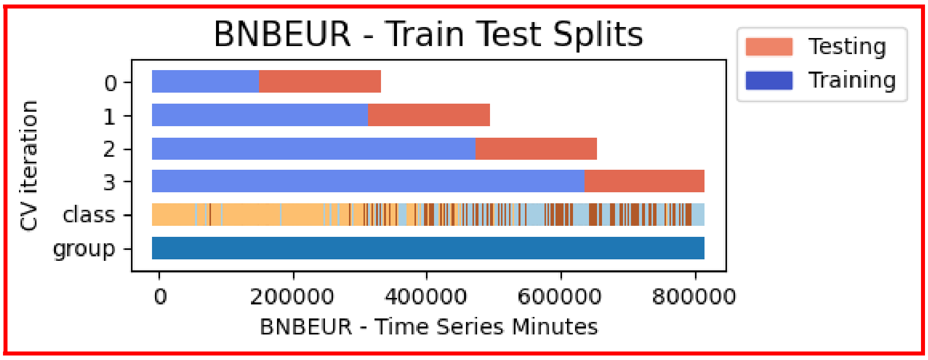


Figure 9: Ethereum, DogeCoin and BinanceCoin Cross Validation Splits

A final note before commencing the experimental process: all batch learning development is implemented using the 'sklearn' Python package (Pedregosa et al. 2011a) with the 'Pandas' DataFrame object once extracted from AWS S3 using the 'boto3' package. All online model development is using the 'river' package (Montiel et al. 2020a). For every batch and online machine learning model, a model alias is provided (eg. B-LG-C).

4.1.1 Exp.1a: Batch Logistic Regression vs Online Softmax Regression

Implementation of Batch Logistic Regression (B-LG-C):

Batch Logistic Regression is developed using sklearn's linear_model.LogisticRegression class, configured for 'multinomial' problems. In order to arrive at a final/best model, this research deployed grid-search parameter tuning for all 3 alt-coins to identify the optimal 'C' tuning across 8 possibilities. The 'C' parameter represents the "inverse of regularization strength" (Pedregosa et al. 2011a), and ultimately controls the degree to which the model is programmed to avoid over fitting to training data. Secondly, data scaling is conducted using the Standard Scaling technique (also sklearn). This transforms each feature to have a mean of 0, with a standard deviation of 1, and is most effective when working with linear models. During grid-search, the settings shown in table 4 were applied:

Table 4: Logistic regression Grid Search Settings

Hyper-parameters:	Settings:
Solver	SAG (Stochastic Average Gradient)
C (Inverse of regularization)	0.0001, 0.0013, 0.0193, 0.268, 3.727, 51.79, 719.68, 10000
Class Weights	None

The 'sag' is noted for performance for larger datasets, therefore it was used for each test within this experiment (Pedregosa et al. 2011a). Finally, the gird-search revealed that inverse regularisation ('C') tuned to 0.0001 (the lowest provided value) was optimal for all Ethereum and BinanceCoin indicating each time it was better to under-fit the model, By contrast, C was best tuned to 51.79 for DoeCoin. In general however, this implementation struggled with algorithmic convergence during grid search (which took 10+ hours). Indeed, providing the model with class weights proved more problematic than omitting class weights (despite significant class imbalance), and as a result, the classes were not balanced by the model to estimate the optimal C value during grid search. Table 5 showcases the logistic regression models finally deployed after grid search:

Table 5: Batch Logistic Regression Final Models

Alt-coin:	C:	Penalty:	Class Weights:
Ethereum	0.0001	l2	Both (with/without inverse proportions)
DogeCoin	51.7947	l2	Both (with/without inverse proportions)
BinanceCoin	0.0001	l2	Both (with/without inverse proportions)

Implementation of Online Softmax Regression (O-LG-C):

On the contrary, river's linear_model.SoftmaxRegression class is used for the online model. Because scikit-learn StandardScaling is incompatible with online learning, data is also scaled with river's StandardScaling class when streaming from S3. The distinction with online standard scaling is that the true distribution is unknown in advance, therefore a "running mean and a running variance" is calculated to scale the data (Montiel et al. 2020a). This implementation also utilised the HardSamplingClassifier class from the river package to increase model performance. This technique is unique to online machine learning which provides the model a window of previously trained samples which were the most difficult to predict (ie. the hardest samples), with an associated probability on how often to retrain on past data and for-go newly streamed data. As shown in table 6, the SoftmaxRegression model is set up the same for each cryptocurrency, with Stochastic Gradient Descent as the optimiser with 0.1 learning rate.

Table 6: Online Softmax Regression Final Models

Alt-coin:	Optimisation:	Hard Sampling Setting:
Ethereum	SGD 0.1	Probability of 0.2, Sample size of 150
DogeCoin	SGD 0.1	Probability of 0.2, Sample size of 150
BinanceCoin	SGD 0.1	Probability of 0.2, Sample size of 150

Exp.1a Evaluation and Results:

Table 7 displays the performance of the batch and online logistic/softmax regression models across each alt-coin. Note that for each batch model, there are 2 sets of results: 1 set for where class weights were provided to the model (ClassBal), and another where no class weights were provided (NoClassBal). For all result tables, the best model per alt-coin is highlighted in bold text:

Table 7: Exp.1a Results Table

Alt-coin:	Model:	Accuracy:	Precision:	Recall:	F1:
Ethereum	B-LG-C (ClassBal)	0.52	0.48	0.52	0.46
Ethereum	B-LG-C (NoClassBal)	0.56	0.53	0.56	0.5
Ethereum	**O-LG-C**	**0.59**	**0.58**	**0.59**	**0.59**
DogeCoin	B-LG-C (ClassBal)	0.39	0.48	0.39	0.35
DogeCoin	B-LG-C (NoClassBal)	0.37	0.49	0.37	0.33
DogeCoin	**O-LG-C**	**0.45**	**0.45**	**0.45**	**0.45**
BinanceCoin	B-LG-C (ClassBal)	0.49	0.55	0.49	0.39
BinanceCoin	B-LG-C (NoClassBal)	0.56	0.55	0.56	0.44
BinanceCoin	**O-LG-C**	**0.61**	**0.61**	**0.61**	**0.61**

Taking these results at face value, the following themes are identified: (a) across each alt-coin and nearly every metric, online learning model outperforms the batch trained models;

(b) for batch models, better performance is found when class balance is not applied; (c) finally, DogeCoin appears to be hardest dataset for predictions. However, a closer look is provided with the following normalised confusion matrices of batch (NoClassBals) and online logistic regression models[2]:

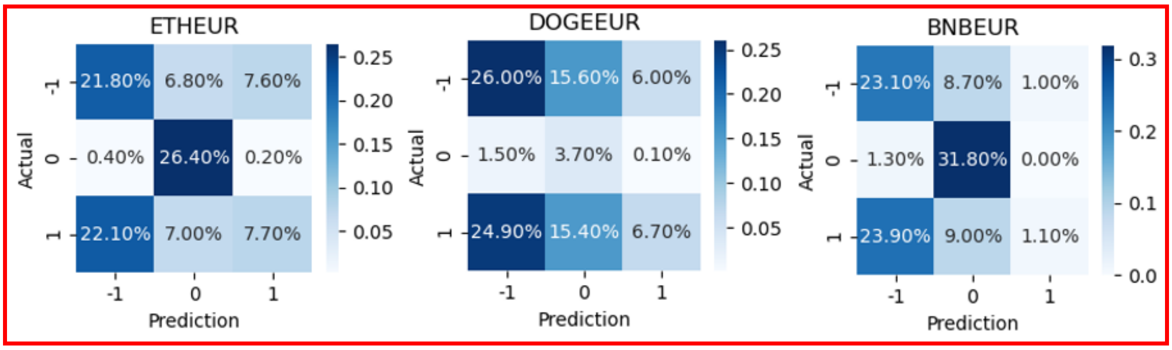


Figure 10: B-LG-C (NoClassBal) Confusion Matrices

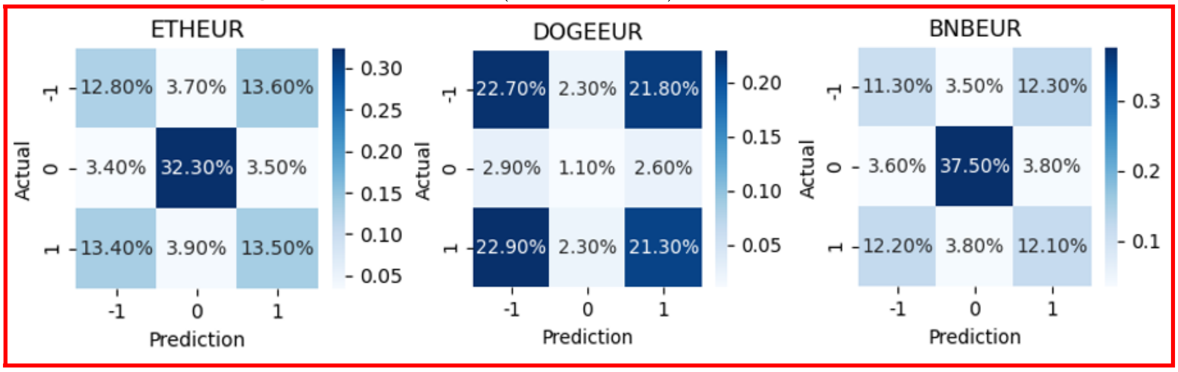


Figure 11: O-LG-C Confusion Matrices

Figure 10 and 11 provide further insight into the model behaviours. For example, the performance variance for BinanceCoin is highly explained by the online models ability to predict '0', which is then compounded by the level of support in the BinanceCoin dataset as outlined in figure 5 and 9 previously. Regardless of these issues, the models are learning from the same data, which in an online manner appears to be preferable. A further inspection of the respective model learning rates (accuracy) is plotted in figure 12:

[2]As described in section 3.1.5, batch model results are averaged across each CV split. Therefore, confusion matrix 'Actual' percentages may not be identical across batch and online evaluations, among other evaluation metrics.

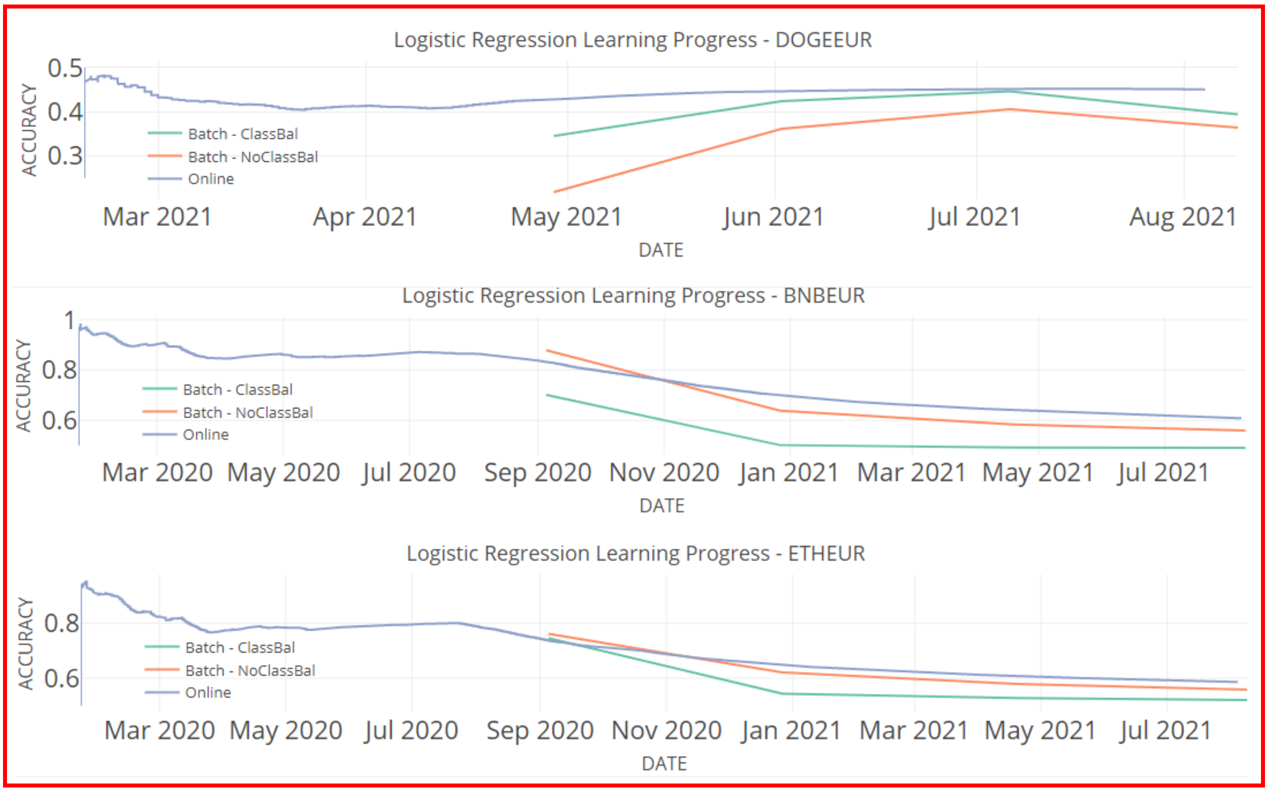


Figure 12: Experiment 1a Learning Progress (B-LG-C vs O-LG-C)

For online models, figure 12 plots the rolling accuracy at every minute. For batch models, there are only 4 data points at the end of each test period, therefore a rolling accuracy is plotted for each split. A close look at the online time series' show the sudden, adaptive learning at the very start of each series, which then tapers downward for Ethereum and BinanceCoin. By contrast, DogeCoin's appears to be trending upward, and with more data could end up with similar performance of the other alt-coins. To conclude, the online implementation is considered preferable, which completes objectives B.1 and C.1 while also partially answering the research question.

4.1.2 Exp.1b: Batch Decision Tree vs Online Hoeffding Tree

For this experiment, a decision tree is compared to a Hoeffding tree (aka. Very Fast Decision Tree (Domingos and Hulten 2000)). The key difference between these two is that a Hoeffding tree can be trained incrementally/online using data streams while decision trees cannot as they require all data in advance (Albert Bifet et al. 2017).

Implementation of Batch Decision Tree (B-DT-C):

For developing the batch decision tree model, sklearn's tree.DecisionTreeClassifier class is used. The sklearn package implements the CART decision tree (Classification and Regression Trees), which builds trees based on the "largest information gain" when learning from the features training the model (Pedregosa et al. 2011a). Because tree-based models do not require feature scaling to be applied (unlike linear models), the original feature values were maintained as opposed to using a scaling technique. For parameter tuning, the grid-search cross validation was applied to establish the best values for the 'criterion', 'splitter' and the maximum tree depth ('max_depth'). For each alt-coin, the time series was split into 4 train-test segments, as shown in figure 9 previously, while table 8 showcases the decision tree grid-search configuration at each train-test split:

Table 8: Decision Tree Grid Search Settings

Hyper-parameters:	Settings:
Criterion	'gini', 'entropy'
Splitter	'best', 'random'
Max depth	5, 10, 20, 30, 40, 50, 60
Class Weights	Inverse proportions of class balance

Accordingly, the best model is recorded at every train-test split, for each alt-coin as shown in the following table. Using this information, the final models were established in figure 9:

Table 9: Batch Decision Tree Final Models

Alt-coin:	Criterion:	Max Depth:	Splitter:	Class Weights:
Ethereum	Entropy	5	Random	Both (with/without inverse proportions)
DogeCoin	Gini	5	Random	Both (with/without inverse proportions)
BinanceCoin	Entropy	5	Random	Both (with/without inverse proportions)

Implementation of Hoeffding Tree (O-HT-C):

This implementation uses river's tree.HoeffdingTreeClassifier class for the online Hoeffding tree classifier. As the data is streamed into all online models, the data is transformed to a Python dictionary, as required by the river package, each record at a time. Because the best parameters are found in batch mode for the decision tree, these are then applied where applicable to the Hoeffding tree. As such, the max tree depth is always 5 for each Hoeffding tree and alt-coin. However, the HardSamplingClassifier is also utilised in this implementation to re-train the model on the fly with the most difficult samples. As per table 10, the settings were as follows for each alt-coin:

Table 10: Online Hoeffding Tree Final Models

Alt-coin:	Criterion:	Max Depth:	Hard Sampling Setting:
Ethereum	Entropy (info_gain)	5	Probability of 0.2, Sample size of 150
DogeCoin	Gini	50	Probability of 0.2, Sample size of 150
BinanceCoin	Gini	5	Probability of 0.2, Sample size of 150

Exp.1b Evaluation and Results:

The performance of the decision and Hoeffding tree are compared herein, which completes objectives B.2 and C.2 while also improving the clarity surround this research question. Accordingly, the results of each model are presented in table 11. Similar to Exp1.a, the batch tree models with no class balancing return higher evaluations. By contrast to Exp.1a however, the performance differential between both batch and online regimes is visibly more difficult to discern. Although O-HT-C's metrics are significantly stronger on the BinanceCoin test, the same cannot be said for Ethereum and Dogecoin. To gain a better intuition about the model outputs, confusion matrices are plotted in figure 13 and 14.

Table 11: Exp.1b Results Table

Alt-coin:	Model:	Accuracy:	Precision:	Recall:	F1:
Ethereum	B-DT-C (ClassBal)	0.5	0.46	0.5	0.39
Ethereum	**B-DT-C (NoClassBal)**	**0.56**	**0.5**	**0.56**	**0.44**
Ethereum	O-HT-C	0.53	0.51	0.53	0.44
DogeCoin	B-DT-C (ClassBal)	0.46	0.46	0.46	0.46
DogeCoin	**B-DT-C (NoClassBal)**	**0.47**	**0.46**	**0.47**	**0.45**
DogeCoin	O-HT-C	0.47	0.44	0.47	0.42
BinanceCoin	B-DT-C (ClassBal)	0.53	0.53	0.53	0.53
BinanceCoin	B-DT-C (NoClassBal)	0.56	0.51	0.56	0.45
BinanceCoin	**O-HT-C**	**0.63**	**0.6**	**0.63**	**0.6**

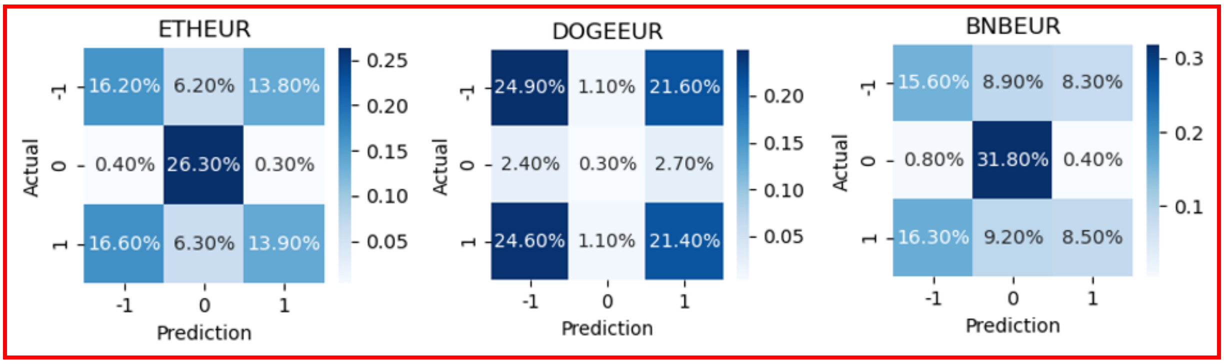


Figure 13: B-DT-C (NoClassBal) Confusion Matrices

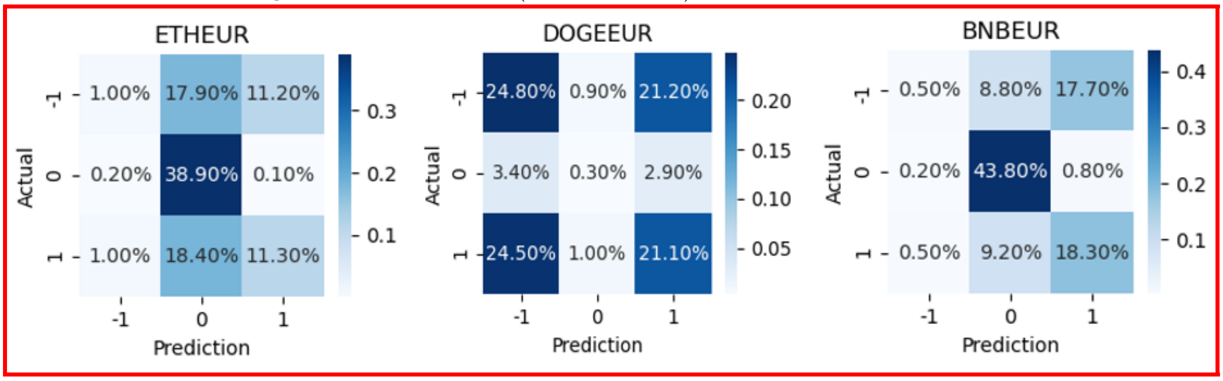


Figure 14: O-HT-C Confusion Matrices

An interesting pattern emerges from figure 14 (for Ethereum and BinanceCoin): each Hoeffding tree was virtually unable to predict the price decreases (-1). Perhaps a minor observation, but this may indicate that a batch trained decision tree offers advantages for multi-class problems. Nonetheless, this certainly contributed to the poorer performance of online models. For the BinanceCoin test, the confusion matrices between B-DT-C and O-HT-C are nearly equally inverted between false price decrease and increase predictions respectively, yet once again, the online model was better able to identify the '0' class for no increase, resulting in a preferable model. Similarly, the learning progression is plotted in figure 15:

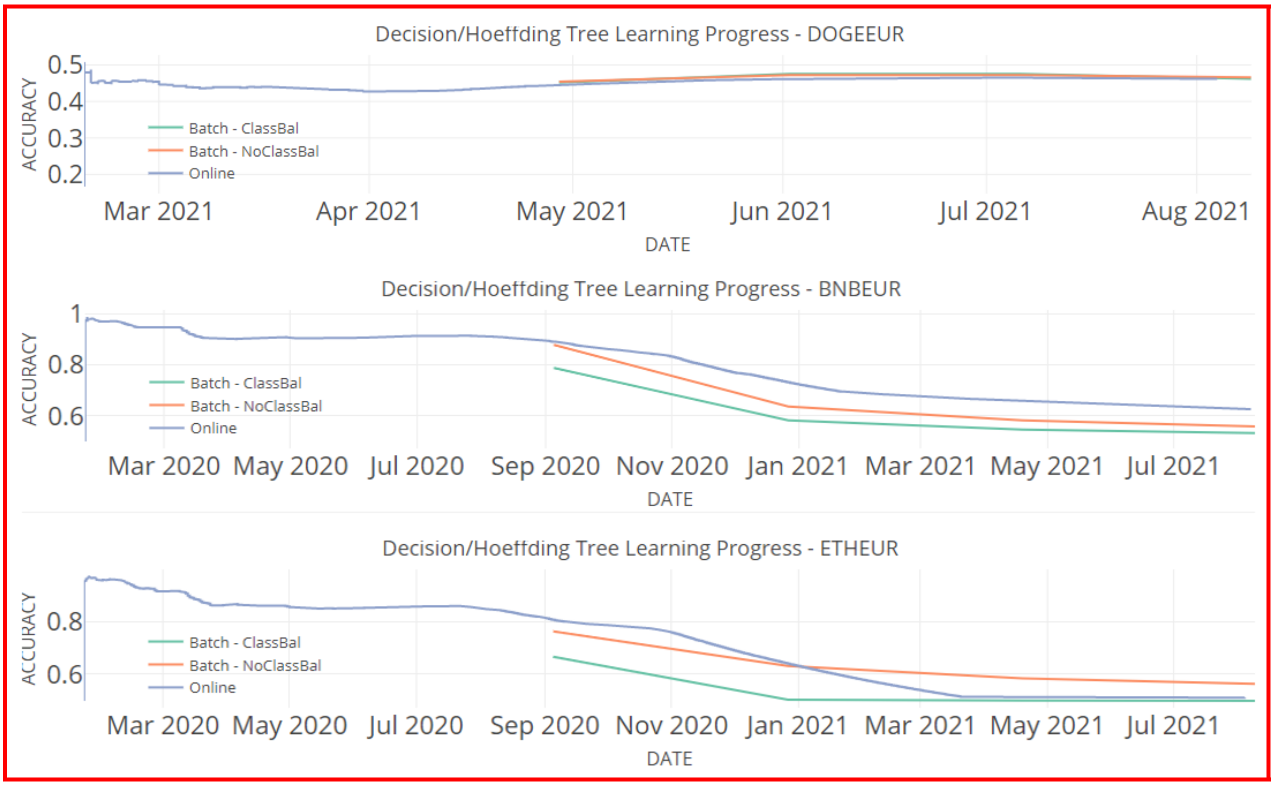


Figure 15: Experiment 1b Learning Progress (B-DT-C vs O-HT-C)

With the benefit of figure 15, it is clear the DogeCoin predictions struggled from start to finish, while Ethereum and BinanceCoin a very clear downward trend emerged, to the point that O-HT-C's learning crossed below B-DT-C after the first training split for the Ethereum test. Because the B-DT-C models outperformed O-HT-C 2 out of 3 datasets, this experiment concludes that batch decision trees may offer minor advantages given that the F1 score for B-DT-C (NoClassBal) showed 0.45, while O-HT-C had 0.42. In general, these results came as surprising due to previous literature noting that Hoeffding trees produce "nearly identical" trees as batch learners when provided enough data Domingos and Hulten (2000), but for alt-coins it appears that different performances are observed, arguably beyond which can be considered "nearly identical".

4.1.3 Exp.1c: Batch Random Forest vs Adaptive Random Forest

Implementation of Batch Random Forest (B-RF-C):

This experiment deploys random forest classification using batch learning as well as online learning, and compares their results. Accordingly, sklearn's ensemble.RandomForestClassifier class is developed for batch learning, which is also deployed using time series cross validation and grid search parameter tuning. However, given the size of the datasets, and the computational overhead required for random forest algorithms, only the following parameters were tested across each alt-coin as shown in table 12:

Table 12: Random Forest Grid Search Settings

Hyper-parameters:	Settings:
Criterion	'entropy'
Number of trees	20, 50
Max depth	5, 10

As always, the goal of grid search is to determine the final random forest model to test in batch and online manners resulting in the models listed in table 13 for each alt-coin:

Table 13: Batch Random Forest Final Models

Alt-coin:	Criterion:	Max Depth:	No.Trees:	Class Weights:
Ethereum	Entropy	5	50	Both (with/without inverse proportions)
DogeCoin	Entropy	10	20	Both (with/without inverse proportions)
BinanceCoin	Entropy	5	50	Both (with/without inverse proportions)

Implementation of Online Random Forest (O-RF-C):

For the online variation, the AdaptiveRandomForestClassifier class is utilised from the river.ensemble module which combines online drift detection with decision trees to identify concept drifts in the data. Upon detecting a drift warning, ”background trees” are trained up which then replace the ”active trees” when concept drift is then fully detected (Montiel et al. 2020a). As per table 14, Adaptive Random Forests deployed were:

Table 14: Adaptive Random Forest Final Models

Alt-coin:	Criterion:	Max Depth:	Hard Sampling Setting:
Ethereum	Entropy (info_gain)	5	Probability of 0.2, Sample size of 150
DogeCoin	Entropy (info_gain)	5	Probability of 0.2, Sample size of 150
BinanceCoin	Entropy (info_gain)	5	Probability of 0.2, Sample size of 1500

Exp.1c Evaluation and Results:

In this experiment, the performance differential between batch and online models is most pronounced with Ethereum and BinanceCoin tests returning notably better performance metrics in favour of online learning. Contrasting this with the DogeCoin test, the lowest performing cyrptocurrency overall, the results show that batch trained random forest models offer slightly preferable results, but this is marginally the case. On aggregate, the adaptive (online) random forest is found to be preferable than a batch trained random forest based on the following results in table 15:

Table 15: Exp.1c Results Table

Alt-coin:	Model:	Accuracy:	Precision:	Recall:	F1:
Ethereum	B-RF-C (ClassBal)	0.49	0.47	0.49	0.41
Ethereum	B-RF-C (NoClassBal)	0.57	0.5	0.57	0.47
Ethereum	**O-RF-C**	**0.63**	**0.61**	**0.63**	**0.62**
DogeCoin	**B-RF-C (ClassBal)**	**0.47**	**0.46**	**0.47**	**0.43**
DogeCoin	B-RF-C (NoClassBal)	0.45	0.47	0.45	0.42
DogeCoin	O-RF-C	0.46	0.45	0.46	0.45
BinanceCoin	B-RF-C (ClassBal)	0.53	0.54	0.53	0.48
BinanceCoin	B-RF-C (NoClassBal)	0.56	0.47	0.56	0.46
BinanceCoin	**O-RF-C**	**0.66**	**0.64**	**0.66**	**0.64**

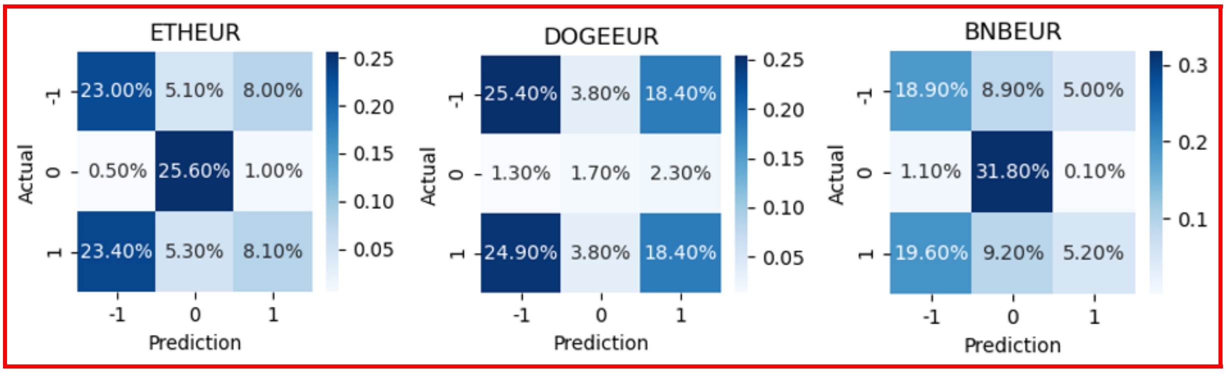


Figure 16: B-RF-C (NoClassBal) Confusion Matrices

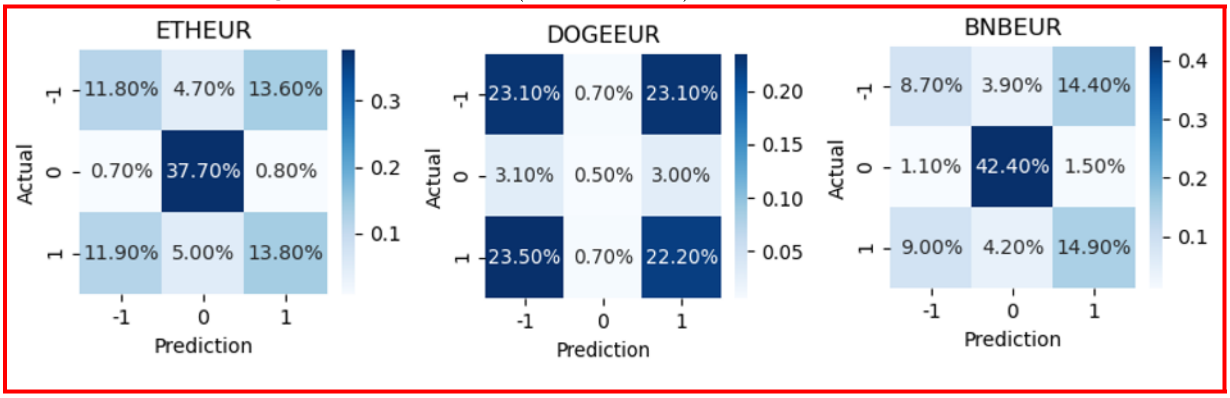


Figure 17: O-RF-C Confusion Matrices

The rationale behind the high performance on the BinanceCoin datasets is once again attributable to extremely high recall of the '0' class, and to a slightly lesser degree, precision too (based figures 16 and 17). Likewise, the learning progression rates are visualised for B-RF-C and O-RF-C in figure 18, which also highlights O-RF-C and B-RF-C's ability to predict within the first 40-50 percent of the time series. However, for the O-RF-C's case, the benefit of the online/adaptive learning becomes very clear in the BinanceCoin dataset between September and January 2021, as the drop rate is not as extreme with online/adaptive learning as it is for B-RF-C, ultimately creating quite a strong performance for O-RF-C on BinanceCoin data (0.66 accuracy). It is also highly promising that an online random forest provides better results overall. This is important due to the computational complexity and costliness of retraining a batch random forest, particularly so for time-sensitive matters such as short term predictions. Objectives B.3 and C.3 are hereby completed, while also completing objective D.1 too.

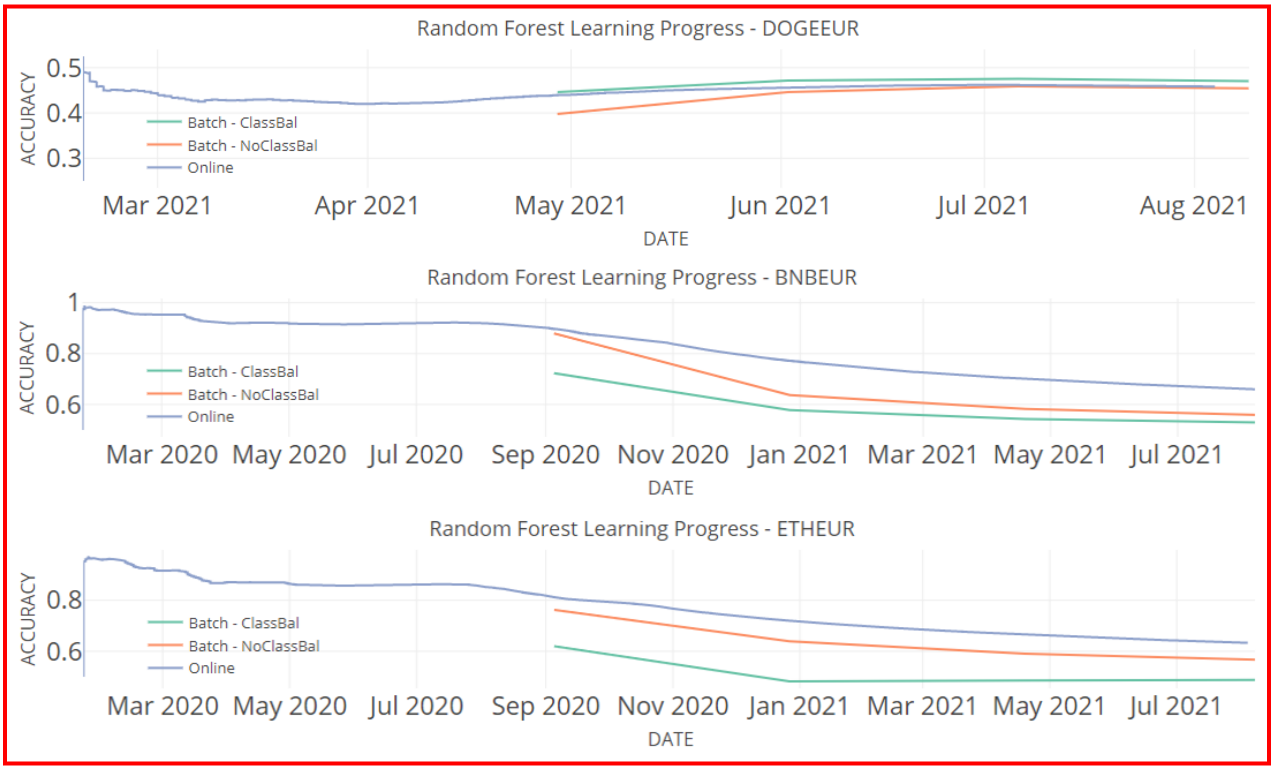


Figure 18: Experiment 1c Learning Progress (B-RF-C vs O-RF-C)

4.2 Experiment 2 Implementation, Evaluation and Results

4.2.1 Implementation of Hypothesis Test

Given the model evaluation metrics collected from Exp1.a-c, the results naturally lended themselves towards performing a 2-sample, unpaired hypothesis test to assist answering the research question in the form of the following hypothesis (significance is 0.05):

> H_0: $\mu_B = \mu_O$: There is no predictive difference between batch and online machine learning models for predicting alt-coin price direction.
>
> H_1: $\mu_B \neq \mu_O$: There is a predictive difference between batch and online machine learning models for predicting alt-coin price direction.

The 2 samples are a collection of all model 'Accuracy' scores, bucketed between batch models (NoClassBal) and online models. The reason all metrics cannot be included is due to the requirement of independent groups, that is, every sample in a group cannot be related to another sample (ie. high accuracy would also result in high F1, etc). This is therefore a relatively small sample size (9 in each set). Due to the absence of normality across each sample (figure 19), an unparametric test was performed: the Wilcoxon Rank Sum test.

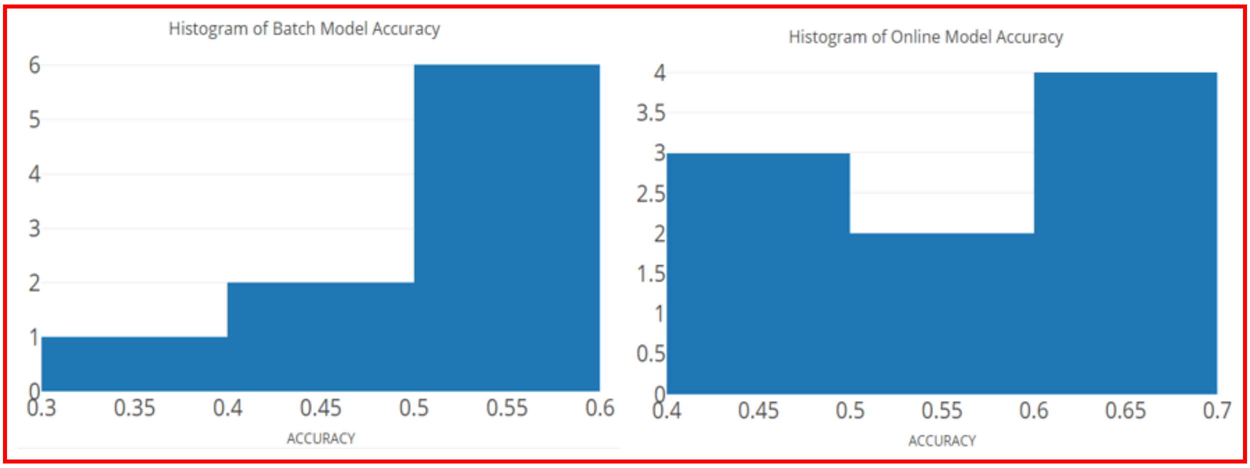


Figure 19: Distribution of Batch (left) and Online (right) model Accuracy

4.2.2 Hypothesis Test Evaluation and Results

Upon running the Wilcoxon Rank Sum test using R (stats package), the p-value returned is 0.23. This indicates that there is not enough evidence provided to confirm that any differences observed by the model performances is not random. Accordingly, the null hypothesis (H_0) failed to be rejected and objective D.2 is completed. Despite this result, one must also interpret this within the business context of predicting alt-coin price direction. That is, even the slightest of performance differences could provide researchers and investors a competitive advantage in cryptocurrency trading/analytics, which may never be identified as statistically significant by a hypothesis test.

4.3 Comparison of Results with Existing Work

This research fits into 2 bodies of literature. First, in the domain of comparing algorithmic performance between batch and online implementations, this research has provided a selection of experiments which while showing mixed results, indicate that online learning offers better performance in 2 out of 3 experiments (Exp1.a, Exp1.c). According to table 16 (and section 2.1), it is clearly difficult to reduce this body of literature into a general consensus on the topic of batch and online learning comparisons strictly when it comes to predictive ability. Nonetheless, the findings of Exp1.c (random forests) are consistent with Jagirdar (2018) who also found online random forests to be preferable.

Table 16: Comparison with Literature - Batch v Online Learning

Reference	Domain	Batch algorithms	Online algorithms	Result (batch/online)
Wang and Han (2014)	Stock time series	Extreme Learning Machine	Online Sequential Extreme Learning Machine, Online SVR	Mixed: online similar predictions less train time
Burlutskiy et al. (2016)	Web-user behaviour	Linear regression, Decision Tree, K-NN, SVM, Deep Belief Network	Stochastic Gradient Descent, Perceptron Neural Network, Passive Aggressive	Mixed: online much faster but slightly less accurate
Liu et al. (2016)	Stock time series	ARIMA	ARIMA-ONS, ARIMA-OGD	Online: better predictions
Suto et al. (2017)	Human activity recognition	Artificial Neural Network, K Nearest Neighbours	Artificial Neural Network, K Nearest Neighbours	Batch: better predictions
Jagirdar, (2018)	Healthcare	Naïve Bayes, Logistic regression, Random Forest, NN, Adaboost	Random Forest, Learn++, Hoeffding Tree	Online: Random Forest accuracy better online
Brenowitz et al. (2020)	Climate	Random Forest, 2-layer Neural Network	Random Forest, 2 layer Neural Network	Batch: better predictions and stability
Barry (2021)	Crypto-currency	Logistic Regression, Decision Tree, Random Forest	Logistic Regression, Hoeffding Tree, Adaptive Random Forest	Mixed: 2/3 tests better results for online (Exp1)

Looking at the cryptocurrency-specific literature, the top results found from experiments 1 (Exp1.a-c) are compared with some of the best results identified throughout literature of classifying future cryptocurrency movement in table 17. Comparing the best results is justified as all studies have a huge amount of results presented:

Table 17: Comparison with Literature - Cryptocurrency Analytics

Reference	Crypto(s)	Dataset	Best Algorithm	Best Results
Amjad and Shah (2017)	Bitcoin	Daily Bitcoin data	ARIMA, *Unnamed classifier*	Accuracy: 0.78
McNally, Roche and Caton (2018)	Bitcoin	Daily Bitcoin data	LSTM	Accuracy: 0.52 Precision: 0.35 Specificity: 0.61
Akyildirim, Goncu and Sensoy (2018)	12 alt-coins	15, 30, 60 minute and daily prices	ANN MLP (out-sample tests)	Accuracy: 0.64 (OMG-Network)
Chen, Narwal and Schultz (2019)	Ethereum	Hourly Ethereum data	ARIMA	Accuracy: 0.61
Albariqi and Winarko (2020)	Bitcoin	Daily Bitcoin data	MLP	Accuracy: 0.81 Precision: 0.81 Recall: 0.94
Sebastiao and Godinho (2021)	Bitcoin, Ethereum, Litecoin	Daily Ethereum data	Random forest	Accuracy: 0.6 (Ethereum)
Jaquart, Dann and Weinhardt (2021)	Bitcoin	Minutely Bitcoin data, Tweets, Financial data	LSTM	Accuracy: 0.52
Barry (2021)	Ethereum, DogeCoin, BinanceCoin	Minutely alt-coin data	Random Forest	Accuracy: 0.66 Precision: 0.64 Recall: 0.66 (BinanceCoin)

Accordingly, it can be seen that the best results found by this research rank reasonably well against other authors. However, this research also had many results which were quite low (below 0.5). The unique element of this best contribution of 0.66 accuracy is that this result was obtained using an online, adaptive random forest model while the others are using batch techniques. By highlighting this contribution, this completed objective D.3.

5 Discussion

The theme to be drawn from the results is that online learning models offers reasonable advantages over batch trained models when classifying next-minute alt-coin price direction. By sampling 3 alt-coins (Ethereum, DogeCoin, BinanceCoin) across 3 batch trained models (Logistic Regression, Decision Tree classifier, Random Forest classifier) and 3 online learning models (Softmax Regression, Hoeffding Tree classifier, Adaptive Random Forest) it is with confidence that these results can be consumed as a useful, informative guide to algorithmic decision making and model development for alt-coin predictions. That being said, the scope of this research has limited remits by only considering a small portion of machine learning models, all of which solved classification tasks, and as such, results should be interpreted within these bounds. To the extent that online learning models offer advantages in the case of SoftMax Regression and Adaptive Random Forests in Exp1.a and Exp1.c (ie. 2/3 experiments in this paper), it is difficult to assert how these findings will generalise to other batch/online learning algorithms and indeed, alt-coins, especially in light of the challenge with the DogeCoin predictions being reliably the toughest to predict overall. In addition, contrasting results between Exp1.a and Exp1.b of the Ethereum dataset indicate the results are too close to be making general claims. This is also buttressed by experiment 2 which indicates that the observed performance differential between models' accuracies were not statistically significant.

It is likely that the following limitations contributed to the lower DogeCoin results: the dataset size was significantly smaller, with only circa 220,000 records, while Ethereum and BinanceCoin were closer to 800,000 records. Furthermore, because DogeCoin was identified as being most volatile in section 3.1.1, these two points are consistent with findings from Liew et al. (2019) who noted that smaller, more volatile alt-coins are harder to predict than the larger, less volatile cryptocurrencies. Notwithstanding this limitation, the research question is not effected by this due to the fact all experiments (Exp1.a-c) use the same 3 datasets, meaning any potential issues with Dogecoin datasets will effect all experiments equally. Frankly though, many batch learning results in this paper do not perform to the same standard found throughout literature (specifically, those under 0.5 accuracy), which may indicate that batch solutions presented in this paper ought to be improved with increased model development, feature engineering, and parameterisation.

Already, a number of alternative avenues come to mind which may have avoided various issues faced in this project, particularly regarding class imbalance. First, focusing on regression tasks to predict the actual price of the alt-coin (despite other challenges which come into play - auto correlation, stationarity, etc) would equally have contributed to answering the research question. Moreover, this research could have benefited from carrying out a greater amount of feature engineering tasks (ie more technical indicators, SMA crossover signals, etc), rather than expecting the models to learn these patterns entirely themselves. On another note, it is often said that not having enough data is a limitation of research problems, but in many ways, the relatively larger sizes of the datasets used in this project certainly contributed to the limited range of grid-searches, tree depths, or ensemble number of estimators being tested, due to the apparent computational complexity. However, the algorithmic efficiency of each implementation was not within the scope of this research, but a greater emphasis on this would also have been a valid inquiry.

6 Conclusion and Future Work

This section concludes this thesis by highlighting the key findings and future work areas with regard to the research question and objectives posed in section 1.1:

Research Question: "To what extent can online machine learning provide an advantage over traditional batch machine learning techniques for researchers and investors when predicting next-minute cryptocurrency (alt-coin) price direction?"

In order to answer this question, the following tasks were completed. First, a comprehensive cloud-based data lake (AWS S3 bucket) of alt-coins priced in Euro was created, thereby allowing data processing for predictive modelling to be possible for both batch and online learning regimes. By deploying machine learning classification techniques using both batch learning (ie. traditional machine learning) as well as the increasingly utilised online learning (ie. incremental or data stream machine learning), machine learning techniques were fairly compared to provide insight of the predictive performance of bringing a model from batch deployment to online within the context of next-minute alt-coin predictions. Secondly, upon collecting all evaluation metrics from Experiment 1, this enabled hypothesis testing to also be completed as outlined in section 4.2. Finally, a cloud-based dashboard is developed for users to inspect results (completing objective E.1), which shows the relative performance over a range of classification tests, indicating that online learning is preferable in 2 out of 3 experiments (Exp1.a, Exp1.c), that is, the online softmax regression model outperformed a batch logistic regression (all 3 alt-coins), and an online adaptive random forest model outperformed the batch random forest (2/3 alt-coin datasets). The online Hoeffding tree did not outperform a batch trained decision tree on aggregate, despite providing better results for 1 of the alt-coins (BinanceCoin).

There a number of future work directions which are still be to explored. First, a natural extension of this implementation would be to identify the performance differential between batch and online regression models for predicting the actual next-minute price of selected alt-coins. Furthermore, given that the overall model performances were not highly promising, it would make sense to also extend this to using deep learning, neural network models such as LSTM among other sequential learning models and other ensemble model types, applied to other less documented alt-coins. Given that this project simulated real-time machine learning using data streams from AWS S3, with future work it would be beneficial to integrate this implementation into a true real-time environment using live Binance API data streams where the models can learn from every minute in real-time.

Acknowledgements

It is with a great deal of gratitude and joy that I now complete the Masters in Science in Data Analytics at the National College of Ireland (NCI). A challenging 2 years, which I could not have completed without the fullest of support from the people I am lucky enough to have around me. First, an enormous thanks to my girlfriend Amy (and our lovely housemate, Wilbur) for all of her support and wisdom throughout the last 2 years, and for putting up with my inconvenient schedules. Lots to look forward to now! Indeed, this also extends to my close family and friends of whom I look forward to spending more time with! Another very special thank you goes to my employer, Fidelity Investments Ireland, and particularly the IO Data Science and Analytics team/management for allowing me this great privilege through their support and encouragement throughout this qualification. Finally, words cannot describe my gratitude to Catherine Mulwa of NCI for over 6 months of tutorship, expertise, and most notably her unrelenting commitment towards top academic standards, through which I have learned an abundance. I thoroughly enjoyed working with you, and all my previous NCI lecturers as a student, and now I look forward to what the future brings.

References

Akyildirim, Erdinc, Ahmet Göncü and Ahmet Sensoy (Nov. 2018). 'Prediction of Cryptocurrency Returns using Machine Learning'. In:

Albariqi, R. and E. Winarko (2020). 'Prediction of Bitcoin Price Change using Neural Networks'. In: *2020 International Conference on Smart Technology and Applications (ICoSTA)*, pp. 1–4. DOI: `10.1109/ICoSTA48221.2020.1570610936`.

Amjad, Muhammad and Devavrat Shah (2017). 'Trading bitcoin and online time series prediction'. In: *NIPS 2016 Time Series Workshop*. PMLR, pp. 1–15.

Bifet, Albert et al. (2017). 'Extremely Fast Decision Tree Mining for Evolving Data Streams'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: Association for Computing Machinery, pp. 1733–1742. ISBN: 9781450348874. DOI: `10.1145/3097983.3098139`. URL: `https://doi.org/10.1145/3097983.3098139`.

Brenowitz, Noah D et al. (2020). 'Machine Learning Climate Model Dynamics: Offline versus Online Performance'. In: *arXiv preprint arXiv:2011.03081*.

Burkov, Andriy (2019). *The Hundred Page Machine Learning Book*. Andriy Burkov.

Burlutskiy, Nikolay et al. (2016). 'An investigation on online versus batch learning in predicting user behaviour'. In: *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, pp. 135–149.

Cao, Hong et al. (2013). 'Integrated Oversampling for Imbalanced Time Series Classification'. In: *IEEE Transactions on Knowledge and Data Engineering* 25.12, pp. 2809–2822. DOI: `10.1109/TKDE.2013.37`.

Chen, Matthew, Neha Narwal and Mila Schultz (2019). 'Predicting price changes in Ethereum'. In: *International Journal on Computer Science and Engineering (IJCSE) ISSN*, pp. 0975–3397.

Chiu, Jonathan and Thorsten V Koeppl (Sept. 2017). 'The Economics of Cryptocurrencies – Bitcoin and Beyond'. In: DOI: `http://dx.doi.org/10.2139/ssrn.3048124`.

Coinmarketcap.com (2021). *Today's Cryptocurrency Prices by Market Cap*. `https://coinmarketcap.com/`. Accessed: 2021-05-05.

Conway, Luke (2021). *The 10 Most Important Cryptocurrencies Other Than Bitcoin*. `https://www.investopedia.com/tech/most-important-cryptocurrencies-other-than-bitcoin/`. Accessed: 2021-05-04.

Domingos, Pedro and Geoff Hulten (2000). 'Mining High-Speed Data Streams'. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '00. Boston, Massachusetts, USA: Association for Computing Machinery, pp. 71–80. ISBN: 1581132336. DOI: `10.1145/347090.347107`. URL: `https://doi.org/10.1145/347090.347107`.

Gomes, Heitor Murilo et al. (2019). 'Machine learning for streaming data: state of the art, challenges, and opportunities'. In: *ACM SIGKDD Explorations Newsletter* 21.2, pp. 6–22.

Guo, T., A. Bifet and N. Antulov-Fantulin (2018). 'Bitcoin Volatility Forecasting with a Glimpse into Buy and Sell Orders'. In: *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 989–994. DOI: 10.1109/ICDM.2018.00123.

Hilbe, Joseph M (2011). 'Logistic Regression.' In: *International encyclopedia of statistical science* 1, pp. 15–32.

Hoi, Steven CH et al. (2018). 'Online learning: A comprehensive survey'. In: *arXiv preprint arXiv:1802.02871*.

Horvat, Nebojša et al. (July 2020). 'Big Data Architecture for Cryptocurrency Real-time Data Processing'. In:

Hossin, Mohammad and MN Sulaiman (2015). 'A review on evaluation metrics for data classification evaluations'. In: *International Journal of Data Mining & Knowledge Management Process* 5.2, p. 1.

Huang, Jing-Zhi, William Huang and Jun Ni (2019). 'Predicting bitcoin returns using high-dimensional technical indicators'. In: *The Journal of Finance and Data Science* 5.3, pp. 140–155. ISSN: 2405-9188. DOI: https://doi.org/10.1016/j.jfds.2018.10.001. URL: https://www.sciencedirect.com/science/article/pii/S2405918818300928.

Islam, Md et al. (July 2018). 'Cryptocurrency vs. Fiat Currency: Architecture, Algorithm, Cashflow Ledger Technology on Emerging Economy'. In: pp. 69–73. DOI: 10.1109/ICT4M.2018.00022.

Jagirdar, Nikhil Mukund (2018). 'Online Machine Learning Algorithms Review and Comparison in Healthcare'. In:

Jaquart, Patrick, David Dann and Christof Weinhardt (2021). 'Short-term bitcoin market prediction via machine learning'. In: *The Journal of Finance and Data Science* 7, pp. 45–66. ISSN: 2405-9188. DOI: https://doi.org/10.1016/j.jfds.2021.03.001. URL: https://www.sciencedirect.com/science/article/pii/S2405918821000027.

Jay, P. et al. (2020). 'Stochastic Neural Networks for Cryptocurrency Price Prediction'. In: *IEEE Access* 8, pp. 82804–82818. DOI: 10.1109/ACCESS.2020.2990659.

Jurafsky Daniel Martin, James H. (2020). *Speech and Language Processing (3rd ed. draft); Chapter 5 - Logistic Regression*. URL: https://web.stanford.edu/~jurafsky/slp3/5.pdf.

LaValley, Michael P (2008). 'Logistic regression'. In: *Circulation* 117.18, pp. 2395–2399.

Liang, N. et al. (2006). 'A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks'. In: *IEEE Transactions on Neural Networks* 17.6, pp. 1411–1423. DOI: 10.1109/TNN.2006.880583.

Liew, Jim et al. (Nov. 2019). 'Cryptocurrency Investing Examined'. In: *The Journal of the British Blockchain Association* 2, pp. 1–12. DOI: 10.31585/jbba-2-2-(2)2019.

Liu, Chenghao et al. (2016). 'Online arima algorithms for time series prediction'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1.

McNally, S., J. Roche and S. Caton (2018). 'Predicting the Price of Bitcoin Using Machine Learning'. In: *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 339–343. DOI: 10.1109/PDP2018.2018.00060.

Mohapatra, S., N. Ahmed and P. Alencar (2019). 'KryptoOracle: A Real-Time Cryptocurrency Price Prediction Platform Using Twitter Sentiments'. In: *2019 IEEE International Conference on Big Data (Big Data)*, pp. 5544–5551. DOI: 10.1109/BigData47090.2019.9006554.

Montiel, Jacob et al. (2020a). *River: machine learning for streaming data in Python*. arXiv: 2012.04740 [cs.LG].

— (2020b). *SoftmaxRegression*. URL: https://riverml.xyz/latest/api/linear-model/SoftmaxRegression/.

Munim, Ziaul Haque, Mohammad Hassan Shakil and Ilan Alon (2019). 'Next-Day Bitcoin Price Forecast'. In: *Journal of Risk and Financial Management* 12.2. ISSN: 1911-8074. DOI: 10.3390/jrfm12020103. URL: https://www.mdpi.com/1911-8074/12/2/103.

Nistrup, Peter (2019). *Retrieving Full Historical Data for Every Cryptocurrency on Binance BitMex Using the Python API*. https://betterprogramming.pub/easiest-way-to-use-the-bitmex-api-with-python-fbf66dc38633.

Pang, Yan, Ganeshkumar Sundararaj and Jiewen Ren (2019). 'Cryptocurrency Price Prediction Using Time Series and Social Sentiment Data'. In: *Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*. BDCAT '19. Auckland, New Zealand: Association for Computing Machinery, pp. 35–41. ISBN: 9781450370165. DOI: 10.1145/3365109.3368785. URL: https://doi.org/10.1145/3365109.3368785.

Pedregosa, F. et al. (2011a). 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

— (2011b). 'Visualizing cross-validation behavior in scikit-learn'. In: URL: https://scikit-learn.org/stable/auto_examples/model_selection/plot_cv_indices.html#sphx-%20glr-auto-examples-model-selection-plot-cv-indices-py.

Rahnama, A. H. A. (2014). 'Distributed real-time sentiment analysis for big data social streams'. In: *2014 International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 789–794. DOI: 10.1109/CoDIT.2014.6996998.

Sabry, F. et al. (2020). 'Cryptocurrencies and Artificial Intelligence: Challenges and Opportunities'. In: *IEEE Access* 8, pp. 175840–175858. DOI: 10.1109/ACCESS.2020.3025211.

Sebastião, Helder and Pedro Godinho (2021). 'Forecasting and trading cryptocurrencies with machine learning under changing market conditions'. In: *Financial Innovation* 7.1, pp. 1–30.

Suto, J. et al. (2017). 'Recognition rate difference between real-time and offline human activity recognition'. In: *2017 International Conference on Internet of Things for the Global Community (IoTGC)*, pp. 1–6. DOI: 10.1109/IoTGC.2017.8008967.

Wang, Xinying and Min Han (2014). 'Online sequential extreme learning machine with kernels for nonstationary time series prediction'. In: *Neurocomputing* 145, pp. 90–97.