1_DataColl2026SysConDRI_AlgTr

by Youry

General metrics

39,991	6,137	428	24 min 32 sec	47 min 12 sec
characters	words	sentences	reading time	speaking time

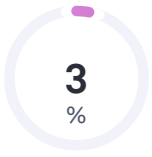
Score

This text scores better than 82% of all texts checked by Grammarly

Writing Issues

219	99	120
Issues left	Critical	Advanced

Plagiarism



31 sources

3% of your text matches 31 sources on the web or in archives of academic publications

Writing Issues

113	**Correctness**	
28	Incorrect punctuation	
7	Text inconsistencies	
18	Confused words	
18	Ungrammatical sentence	
4	Faulty subject-verb agreement	
1	Mixed dialects of english	
8	Misspelled words	
2	Misplaced words or phrases	
2	Pronoun use	
3	Incorrect verb forms	
4	Closing punctuation	
4	Improper formatting	
1	Conjunction use	
1	Incorrect phrasing	
9	Determiner use (a/an/the/this, etc.)	
2	Wrong or missing prepositions	
1	Incorrect noun number	
93	**Clarity**	
84	Paragraph can be improved	
7	Intricate text	
2	Wordy sentences	
6	**Engagement**	
6	Word choice	

7	**Delivery**
4	Inappropriate colloquialisms
1	Incomplete sentences
1	Potentially sensitive language
1	Tone suggestions

Unique Words

Measures vocabulary diversity by calculating the percentage of words used only once in your document

25%

unique words

Rare Words

Measures depth of vocabulary by identifying words that are not among the 5,000 most common English words.

41%

rare words

Word Length

Measures average word length

5.1

characters per word

Sentence Length

Measures average sentence length

14.3

words per sentence

1_DataColl2026SysConDRI_AlgTr

Algorithmic Trading Subsystems Data Collection and Transformation Process Automation Using DRI for Machine Learning Modelling and Forecasting

Justin Drenka
Computer Science
UBCO
Kelowna, Canada
0009-0008-6821-8379
Helana Jaraiseh
Computing Science
UFV
Abbotsford, Canada
0009-0000-1239-2820

James Midtdal
Computer Science
Okanagan College
Kelowna, Canada
0009-0005-6312-569X
Kristina Cormier
Computer Science

Okanagan College

Kelowna, Canada

0009-0004-3783-9704

Youry Khmelevsky

Computer Science

Okanagan College

Kelowna, Canada

0000-0002-6837-3490

Gaetan Hains[2]

LACL

Universite Paris-Est´

Creteil, France[2]

0000-0002-1687-8091

Albert Wong

Mathematics and Statistics

Langara College

Vancouver, Canada

0000-0002-0669-4352

Brandon Hay

Computer Science

Okanagan College

Kelowna, Canada

0009-0001-6605-6037

Abstract—This research focuses on the development of an automated data collection and transformation framework for Algorithmic Trading (AT) predictions using machine learning (ML). The implementation of our computational resources is supported by the Digital Research Alliance of Canada (DRAC), Alliance Cloud Connect Pilot. This enables largescale data processing, storage, and model training. Historical market data and real-time data streams are collected at 5minute intervals and stored in staging tables and will be aggregated into our model table as 15-min interval data. It is important to ensure that data from multiple sources remain consistent, accurate, and can be reliably reproduced throughout the data pipeline.

Index Terms—Algorithmic Trading, Machine Learning
Modelling, System Integration, Performance Optimization

I. INTRODUCTION

Youry

maybe Krsitina

This paper discusses some of the solutions to acquiring complete historical and real-time data sets, along with the significant resources needed for storage and computation, in order to train a machine learning model for short-term Algorithmic Trading predictions. The focus during this phase of the project is on data collection and transformation. The next phase will cover data modeling. For our last phase, we will explore training and testing our machine learning model.

The Alliance Cloud Connect Pilot allocates 200 coreyears on the NIBI-compute system, 20.0 RGU-years on the nibi-gpu system, and 59 TB of project storage on the NIBI storage system for High Performance Computing (HPC). It also

allocates 16 VCPU-years, 4 Number of cloud instances, 32 GB of RAM, 7 volumes, 7 snapshots, 2 Floating IP addresses, 60,000 GB of cloud volume and snapshot storage of Cloud allocations on the Arbutuspersistent-cloud system. After a project is approved and the resources are allocated a few more steps need to be taken to access these resources. One of the first things we needed to do was set up a Secure Shell (SSH) key to access our resources. Another important step was to set up Multifactor Authentication (MFA), which is used when we sign into DRAC and when you are using the SSH tunnel to access our server. Furthermore, users will have to send an email to DRAC, requesting access to certain resources and specifying the quantity required, and wait for a reply before they can create and connect to their PostgreSQL database.

YK: new paper contribution

Helana: new API to the DBMS testing howto train and test data and explain in section IV – Justin: Please work with Section III to explain everything, including design, jobs, access to the servers, and log testing – James: Please test everything and add comments or requests to Justin related to his part. – ALL: Keep in mind, we can reuse 25% of the previous paper. Please comment out the used text.

With these resources, we are now able to collect historical data for the past thirty years. Our previous work only included the three previous years. Our new data set will be ten times larger than our previous set.

One of our biggest challenges is related to the fact that many of the previous sets of data are incomplete and they are leading to unexpected results with the machine learning model. We are working on patching the missing values and increasing the size of our data set.

The main contributions of this paper are (1) a new approach to data collection from external data sources for XGBoost training, testing, and stock forecasting;

(2) the new and historical datasets uploaded to an Oracle/PostgreSQL general purpose DBMS within DRI research project; and (3) the design and testing of a new API for direct access to the DBMS from the core subsystem as part of machine learning (ML) training, testing, and forecasting on the vast datasets collected, transformed, and stored in the DBMS.

II. EXISTING WORKS (FROM OLD PAPER FOR NOW)

Algorithmic trading systems and ML algorithms for predicting stock price movements have recently gained popularity. Much of the research in this area, including studies by Al-Akashi and Hassan [1], Kolte et al. [2], and Li [3], has focused on refining statistical learning methods to improve prediction accuracy. At the same time, relatively limited attention has been given to the quality and reliability of the data pipeline.

On the other hand, there has been a concerning neglect in developing an algorithmic trading system that is operationally efficient [4]. Data is an essential component of a successful algorithmic trading system. However, The main issues remain in acquiring, transforming, and storing the large datasets required for such a system [5]. Many algorithmic trading and ML forecasting projects were limited by the size of the data set in model development and by the lack of system support in their possible implementation of the developed models. As Theate and Damien [6]´ pointed out, training of their ML models is wholly based on "generating artificial trajectories from a limited stock market historical data set." Therefore, the need to develop a data depository, such as a DW, to support an algorithm trading and stock prices forecasting project is apparent.

For ML model development, the required data for training and testing are collected from different sources. Yulianto concludes in their research [7] that the heterogeneity of data from various sources can be dealt with by designing

an "Extract, Clean, Conform, and Delivery/Load" process. Capturing heterogeneous data from different sources and storing it in single or multiple data depositories is a common issue for many organizations. In addition, the diversity of database structures is also a big challenge. Azeroual et al. [8] concluded that implementing an ETL process for data filtering, cleaning, verification, and aggregation could overcome these challenges.

This emerging emphasis on leveraging modern technologies' computational and data-handling capabilities was also articulated in Oyewale et al. [9]. The paper states that market complexity and volatility, along with the volume and speed at which financial data is generated, shape the challenges surrounding stock market analysis. These challenges underscore the need for a system that can handle large datasets in a way that supports accurate realtime predictions in a high-frequency trading environment.

According to Haryono et al., [10], ETL and ELT (Extract/Load/Transform) are the primary data processing methods for implementing a DW. Choosing the correct method is problematic because consideration of a company's cost, efficiency, and procedure plays a vital role in determining the implementation of a DW.

Furthermore, Katari and Rodwal [11] in their paper discuss the inadequacy of traditional ETL processes in handling the volume and inconsistency of the data inherent to financial markets. Their paper outlines the integration of three automated ETL processes that use AI and ML to handle data similar to what we will deal with. They state that harnessing the capability of AI and ML techniques had advantages relating to enhanced accuracy, scalability, speed, and integration, among other benefits.

Patel et al. compared the features of different tools in their literature review [12]. They outline the development of various categories of ETL tools, some of

them are "code-based, GUI-based, cloud-based, Metadata support, Real-time support, and batch processing".

The efficiency of the ETL System is a high priority. According to Biswas et al., [13], many organizations use commercially available, GUI-based commercial products as the ETL solution. However, they compared the performance of four code-based ETL tools, Pygrametl, Petl, Scriptella, and ETL, and discovered that custom-coded tools provide better performance and efficiency in specific cases.

A prevailing theme that emerges from our research is the critical role of data quality and quantity in generating accurate stock price predictions. However, conceptualization and analyses of systems capable of generating such data pipelines are scarce in the literature. Despite the known importance of data quality, relatively few studies investigate the complete life cycle of data, from collection through ETL processes to the DW and into ML algorithms for forecasting in real-time [14].

In the papers we reviewed, we also discovered a lack of use of cloud resources that severely limits the scalability and flexibility of implementing a data depository. Cloudbased ETL systems are considered superior as they provide higher scalability and flexibility for the growing dataset

[?].

Recent research projects have also explored several novel ETL implementation techniques. Wu et al. [15] propose a system using Apache Airflow in combination with Python scripts to orchestrate and automate ETL tasks. Their framework demonstrates the advantages of scheduling ETL processes at regular intervals, automatically extracting data from financial sources, transforming it to fit a standardized schema, and loading it into a custom DW. While this model operates on a smaller scale and updates less frequently than

high-frequency trading systems require, it highlights the benefits of automation for consistency, error reduction, and efficiency in data loading.[53] Our previous analyses on several implementation techniques were documented in [?].

NGOC-BAO-VAN et al. [?] discussed designing[54] and implementing[54] a coffee commodity trading DW[54] to support informed decision-making. After data extraction, the ETL process extracts,[55] cleans,[55] and transforms the[55] data from multiple sources. Visualization and analysis for price trends[56] are performed. The current system could handle structured data successfully but[57] lacked integration with cloud resources, limiting its flexibility and scalability.

Based on these observations, we choose an objectoriented[58] approach, a well-established and proven method for creating compelling and robust systems, to design an efficient ETL process in this research project. We also use Python scripts, PostgreSQL DW, and a user interface (UI) to improve the performance of the ETL process.

III. A NEW APPROACH FOR DATA COLLECTION AUTOMATION USING DRI (JUSTIN AND MINAMI)

A. Overview of the Auto Data Collector System

In order to have[59] accurate short term[59,60] forecasting, the model[52] needs to always have knowledge of recent market activity.[61,62] The automated data collection system is designed to provide this by continuously updating the model dataset with current market information.

B. Cloud Deployment and Scheduling

The automated collector lives on a virtual machine instance within the Arbutus Cloud, part of Canada's[2] Digital Research Infrastructure (DRI). The system runs on a p4-6gb[63] instance, which provides 4 virtual CPUs, 6 GB of RAM, and a 20 GB persistent disk. Hosting the process in the cloud[64] allows for consistent uptime,

simple remote access, and protection against local hardware or power failures. Arbutus Cloud instances offer persistent storage, allowing the collector to run continuously without manual intervention needed.[65] After about fifteen days of collection for 500 tickers, the PostgreSQL database uses roughly 100 MB of space, most of which is initial PostgreSQL table and index metadata rather than data records.[66] This shows that the available storage is more than enough to support years of collected market data.

Job scheduling is managed using systemd timers, which run data collection at fixed intervals during standard market hours. This scheduling method ensures no data is missed,[67] and reliable execution in the event of a network or system interruption.[68] Each run creates a status log, forming a history of performance for easy review and debugging. This combination of stable cloud infrastructure with automated scheduling,[69] supports a reliable data pipeline that stays synchronized with current market data and requires minimal operator oversight.

As shown in Listing 1, a systemd timer deployed on the VM triggers the collector throughout standard stock market hours.

Listing 1. Systemd timer for the Auto Data Collector

```
[Unit]
Description=Auto Data Collector Timer
# Run data collector twice per market hour
# Time in UTC on the VM
[Timer][70]
OnCalendar=Mon..Fri *-*-* 13:58 # Market open 09:30 EST
OnCalendar=Mon..Fri *-*-* 14:30 OnCalendar=Mon..Fri *-*-* 14:58 ...
OnCalendar=Mon..Fri *-*-* 20:58 # Market closed 16:00 EST
OnCalendar=Mon..Fri *-*-* 21:30
```

Persistent=true

AccuracySec=1s

[Install]

WantedBy=timers.target

The timer triggers the service shown in Listing 2, which executes the collector script.

Listing 2. Systemd service that runs the Auto Data Collector

[Unit]

Description=AutoDataCollector Service

Ensure network is ready before running

After=network.target

[Service]

Run the script once per timer trigger

Type=oneshot

User=almalinux

WorkingDirectory=.../AutoDataCollector/

Entry script executed by the service

ExecStart=.../bin/run_collector.sh

Retry if the script exits with an error

Restart=on-failure

[Install]

WantedBy=multi-user.target

C. Data Collection Process

Fig. 1. Automated Data Collector Architecture and Workflow Design

each execution, a time window is created which covers the most recent trading period, with a small overlap from the previous run to ensure no data are missed.

As shown in Listing 3, the script computes a rolling one-hour EST window that

ends twenty minutes before the current time, creating deliberate overlap between runs to ensure that no time interval is ever missed[74]

Listing 3. Time window calculation used by the Auto Data Collector

Determine rolling 1 hour[75] time window def get_current_time_window():

now_ny =

database.[76] After retrieving recent data from FMP, the results are written to the database with[77] an "[2]upsert"[2] operation, where[77] entries are updated when[77] already present,[77] and inserted[77] when not present yet.[77] Listing 4 shows the portion of the collector that filters each record into the active time window and writes it to the corresponding ticker table using[78] an UPSERT operation.[78]

Listing 4. Filtering and UPSERT insertion of 5-minute records

Fetch, filter, and upsert 5 minute[79] market data into the staging table

rows = [] for record in data:

ts = parse_fmp_timestamp(record["date"]) if window_start <= ts < window_end and

record.[80]get("[2]close")[2] is not None:

datetime.now(ZoneInfo("America/New_York"))

The automated data collection process is handled by a Python script that runs on the virtual machine described in Section B. The collection script is designed to be run continuously throughout standard trading hours. During # End window 20 minutes before now to ensure no missed rows from API

end_time = now_ny.replace(second=0,

microsecond=0) timedelta(minutes=20)

start_time = end_time timedelta(hours=1)

Store timestamps in EST return[81] start_time.replace(tzinfo=None),

end_time.replace(tzinfo=None)

Data is extracted from the FMP API using this time window and the schedule is timed so that each collection happens shortly after new data is available. This keeps the database aligned with recent market activity with a slight lag.

The script uses lists of tickers so that symbols we want to track could be added or removed at any time and the collector would adjust. This makes the system flexible and easier to maintain as data requirements evolve. Because FMP provides timestamps in EST, the script preserves this format when inserting records into the PostgreSQL

```
rows.append((
ts, record["open"], record["high"], record["low"], record["close"],
record["volume"]
))
sql = f"""
INSERT INTO {table} (ts, open, high, low, close, volume)
VALUES %s
ON CONFLICT (ts) DO UPDATE SET open=EXCLUDED.open,
high=EXCLUDED.high, low=EXCLUDED.low,
close=EXCLUDED.close,
volume=EXCLUDED.volume;
"""
# Bulk insert all row tuples into DB execute_values(cursor, sql, rows)
```

Running this process with systemd timers on the cloud based VM ensures a stable environment for collection, allowing the historical dataset to always be up to date, accurate, and ready for training the forecasting model.

D. Database Storage Architecture

The auto collector stores new market data in a set of staging tables in the database, with each ticker getting its own table. Each table uses OHLCV format

(open, high, low, close, and volume), matching the structure provided by our data source (FMP API). These values describe how the market price moved during each time interval, and using the same format as the historical dataset keeps both sources consistent.

Every entry in the staging tables represents one 5 minute interval and is identified by its timestamp. Listing 5 shows the structure of a typical staging table, where the timestamp acts as the primary key and ensures that overlapping collection windows do not introduce duplicates.

Listing 5. Example staging table used for a single ticker

```
-- Staging table for a single ticker (AAPL)
CREATE TABLE IF NOT EXISTS market.aapl ( ts TIMESTAMP PRIMARY KEY, open DOUBLE PRECISION, high DOUBLE PRECISION, low DOUBLE PRECISION, close DOUBLE PRECISION, volume BIGINT
);
```

Choosing the timestamp as the primary key allows the collector to easily update rows when collection windows overlap, and separating each ticker into its own table avoids write conflicts if many symbols were to be updated at the same time. This simple layout makes the storage layer easy to maintain and reliable for continuous data collection.

The staging tables serve as the first stop for new data. Their role is to hold recent market information in the same structure as the historical dataset. In a later step of the pipeline, both the historical data and the auto collected staging data are combined into a single model table used for training the forecasting model. That merging process is described in section: todo: add correct section here

E. Testing and Logging

Testing and logging are important parts of making sure the auto collector captures 100% of the expected market data for each market day. The system includes a daily coverage test that runs automatically about one hour after the market closes. Its job is to check that every 5 minute interval during standard trading hours was collected for every ticker. Listing 6 shows the core logic of the daily coverage test, which verifies that the expected number of five minute intervals was collected for each ticker.

Listing 6. Core logic of the daily coverage test

```
# Single ticker coverage between two dates def coverage_for_symbol(conn,
symbol,
start_date, end_date):
expected_per_day = expected_intervals_per_day()
total_expected = 0 total_actual = 0
table = format_table_name(symbol) current = start_date
with conn.cursor() as cur:
while current <= end_date: # Skip weekends if current.weekday() >= 5: current
+= timedelta(days=1) continue
# Intervals for one trading day total_expected += expected_per_day
# Count found rows for that day open_dt =
datetime.combine(current, time(9,30)).replace(tzinfo=None)
close_dt =
datetime.combine(current, time(16,0)).replace(tzinfo=None)
cur.execute(
f"SELECT COUNT(*) FROM {table}
WHERE ts >= %s AND ts < %s",
(open_dt, close_dt),
) total_actual += cur.fetchone()[0] current += timedelta(days=1)
```

Return coverage percentage return (total_actual / total_expected)

* 100 if total_expected else 0

The test knows the start date of collection and adjusts its expectations as the days go on, so it continuously logs a coverage percentage for each ticker. This helped us catch several issues early, such as missing intervals, FMP API changes, and unexpected execution times. Finding these problems quickly made it much easier for us to ensure the dataset is complete and clean.

In addition to the coverage test, the collector writes a status log for every scheduled run, noting when the job started, when it finished, how long it took and whether any errors occurred. The coverage test also records its results in the logs. Together, these tests and logs provide essential visibility into the system's performance and ensure its data remains accurate over time.

IV. API ARCHITECTURE, DESIGN AND TESTING TO DRI DBMS

A. Machine Learning Training and Testing setup to run against of DRI Database

– Helana and Kristina, please try to connect to database and run on Jupyter Notebook or Hub – and explain how it was done here. You can add a diagram

B. API Architecture, Design and Testing to DRI DBMS

– Helana: new API to the DBMS testing how to train and test data and explain in section IV

Follow ELT Process, instead of ETL.

From Financial Model Prep to Staging Tables: Extract raw stock, index, bond, and commodity data from Financial Model Prep API. Load raw data to individual staging tables. Used automatic collector described in Section III.

From Staging Tables to Model Table: Extract, transform, and merge data and store in model table. Used a Jupyter Notebook.

From Model Table to Model / User Interface: Connect Model to Table; the Model will query Table to fetch training data. User Interface will be built in the future.

From From Staging Tables to Model Table in Detail: A Jupyter Notebook will extract the raw data from the staging tables, merge and transform the data and load them into a single table for model training.

Data Cleaning: FMP's raw data contains missing entries and duplicates. The Jupyter Notebook script creates a dataframe for each dataset, removes duplicates, and uses an *** appropriate imputation method (needs further research) to fill in the missing time entries.

Data Formatting: The extracted data must be formatted to align with the DW schema, including proper conversion of date formats, numerical precision, and standardization of measurement units (e.g., currency, percentages).

Data Transformation: During this process, only categorical values are transformed. Numerical values are normalized during the model training process.

The ETL subsystem was designed to allow us to use multiple endpoints from the FMP API to fetch the required data and fit it into our DW schema. The ETL process design is described in Fig. 2.

V. THE AUTOMATION PROCESS IMPLEMENTATION

– James, please try to describe here the automation process implementation, and even add the code

Fir's crontab tool has been disabled and Fir has minimum requirements on job length. The virtual machine on Arbutus Cloud provides access to scheduled tasks, which is necessary to run our scripts at regular intervals. The disjoint databases and short runtime lengths of many of our scripts require some of the work to be done on Arbutus Cloud and on local (personal) machines, and thus some form of synchronization is required. Therefore, we must automate the process of updating Fir with current stock information.

A. Historical Stock Data Collection

The FMP API, configured to retrieve stock information across 15 minute intraday intervals, will return a maximum of 10 days of data for one stock symbol, per API call. Due to the large number of API calls required, a script is needed to automate data collection across the desired date range. Our approach is to run a Python script on our local machine. We begin by defining a start date, end date, and stock symbols. Then, for each stock symbol, we have an inner loop which iterates through the target range calling the API with distinct 5-day windows. Each time the inner loop terminates, the per-stock results are saved locally into a CSV file named with the corresponding stock symbol. The outer loop terminates when all stock symbols have been processed. The resulting CSV files become input for a script which checks for missing entries. This Python script loads each file based on the stock symbols defined in the data collector. Exclusion dates are inserted into a list (holidays, etc.) so as to not trigger false alerts. The start date, end date, opening time, and closing time are also defined, matching the definitions in the historical data collection script. Expected 5-minute intervals are calculated, then actual results are then compared against the generated list. Any missing values are logged to the console for the operator to investigate. To confirm the API results were correct, missing entries are double-checked with a manual API call. Occasionally, FMP does not have data for an interval but it is critical to validate the missing data is due to FMP's data collection process and not a connection issue. The raw data CSV's are now ready to be patched (missing entries filled with previous entry's information) and/or uploaded to the database.

B. CSV File to Database Upload Automation - Brandon

Fig. 2. Automated ELT Architecture of the Algorithmic Trading System

The CSV stock data can now be inserted into our Fir hosted database using automated python scripts. Before attempting to insert data into the database an SSH Tunnel needs to be made to Fir so the database is accessible to us. The first stock CSV file is read into memory and is cast into the appropriate variable format for the database. As using multiple INSERT commands can take over 10 minutes per stock due to overhead, we take advantage of PostgreSQL's COPY command which acts as a bulk insert and takes approximately 5 seconds per stock. The script then iterates throught the rest of the stock files until all data has been copied into their respective tables.

C. Current Stock Data Collection

In order to update the database with the latest stock information, the FMP API must be called continually throughout the day. The virtual machine on Arbutus Cloud, running a Bash shell on AlmaLinux, provides the mechanism through which our Python script is automated. A systemd timer is configured to run every 30 minutes, covering 4:00 am to 8:00 pm Eastern Standard Time, Monday through Friday. The timer's service executes a Bash script, which performs the following activities in this order: activate the Python virtual environment, initialize environment variables (API key and database connection), run the Python data collection script described in Section III.

D. Database Synchronization (Partially Implemented)

We began by installing firewalld to manage port access. After opening the default PostgreSQL port, we generated a public and private SSH key pair and saved them locally on the Arbutus VM. Next, we used ssh-copy-id to copy our public SSH key to Fir. We are then able to open a secure SSH tunnel to the Fir database, using a non-dedicated port on the Arbutus VM as our local port. Finally, once the port-forwarding is setup is complete, we can use the Arbutus

terminal to run psql using the now open ssh tunnel. This completes the connection and we have command line access to the PostgreSQL database on Fir.

A downside to this method is the two-factor-authentication required for every Fir database access. Future work for this section include discussions with the Digital Research Alliance of Canada to resolve the two-factor-authentication issue. Preferably, we would like to connect Arbutus to Fir exclusively through internal DRAC tools or networks, as opposed to SHH which always requires Duo authentication. The script to automatically insert rows has not yet been written, but database access has been established which opens the possibility of fully automated database synchronization.

E. Transformation - Brandon

The raw data in the staging area must be transformed into a single table, appropriate for the ML model. This table can only consist of numerical data as machine learning models are highly mathematical in nature. The number of stocks in the staging area requires a repeatable transformation process.

The raw data will be queried from the database and sent to Arbutus Cloud for Processing with a Python script. Although processing could be done on the Fir database using PL/pgSQL, it would be much harder to accomplish without the extensive python libraries that make data transformation easy. These libraries include pandas, holidays, and yfinance. First, sector data is queried from yfinance for each stock and stored in a table. Each stocks data is read in from the table and can now be joined to the sector data using pandas. Now commodity, bond and index data can be read in and joined to each individual stock on the date column. The holidays library allows us to retrieve a list of holiday dates so we track the holidays to see how pre and post holidays

Fig. 3. Integration of Automated Data Collection Scripts

VI. DW API DESIGN FOR ML TRAINING AND TESTING

– James and Helana, At first we need to test API access to database from Jupyter Notebook or Hub

Currently, the DW only contains data for 2021 to 2024, but the FMP offers a historical dataset spanning 30 years. Ideally, the DW should store the entire dataset. Estimating the DW's storage requirements for this expansion is critical to identifying storage capacity, query response times, and cost management. Furthermore, protecting the space needed to accommodate ten years of data records is essential. To estimate these requirements, the total number of records in the fact table for forty years of data must be calculated, along with the average size of a single record in the fact table. Based on the hourly grain of the fact table and the two additional records for AM and PM aggregations, 26 records are generated daily for each stock, index, and commodity [?].

A. DW API design

With the integration of the ETL process, historical and current financial data is stored in the DW. After that, it is necessary to allow the other subsystems to access the required data seamlessly. For this purpose, we have developed an API which establishes a connection with the DW using SQLAlchemy [?]. Along with the DW connection, API routes created using Flask provide access to required data types through their respective routes, such as stocks, financial indexes, and commodities. The core subsystem uses this API to query data from the DW based on controls provided on the web UI to filter data based on requirements. The queried data is then presented on the web UI with the help of graphs as shown in Fig. 4.

B. Forecast Reporting

Fig. 4 shows the comparison between the red line, which represents the actual stock prices, and the blue line, indicating the predicted prices over time, provides an analysis of the model's accuracy, where periods of close alignment between the lines suggest successful prediction, while deviations indicate areas for potential improvement. This visualization highlights the model's effectiveness in capturing general stock price movement trends while underscoring challenges in predicting specific fluctuations accurately.

Fig. 4. An Example of DRI DBMS and ML Integration: Forecast Report for Analysis of Stock Prices of Apple, Tesla, and Microsoft

The underlying hardware infrastructure significantly influences the performance of each subsystem. For example, storage drives with high Input/Output Operations Per Second (IOPS) can drastically improve the access time to the DW [?]. In contrast, GPUs with higher CUDA cores accelerate ML training and prediction tasks [?]. In the project's current phase, we are utilizing the hardware resources available within our Computer Science Department, which are sufficient for our limited-scale deployment. However, as the system is migrated to the Cloud, we will be able to scale both the volume of data processed and the frequency of predictions, leading to further optimizations and improvements in performance.

We re-designed the DW, implementing a star schema for efficient data querying . This schema is tailored to the needs of our ML algorithm, ensuring that the data is preaggregated and ready for use. This process eliminates the need for real-time aggregation during model training, as all required aggregations are handled during the ETL process, and the data is ready for consumption as a training dataset. This design reduces the need to join multiple data from tables and improves the overall efficiency of the ETL process, leading to a simpler API

to provide data access for ML model training. The updated workflow can be seen in Fig. 5, modelled after the work from [?].

FUTURE WORKS

In future research, we will use 30 years of financial data for the training and testing models. DRA provides production performance training and offers an array of cuttingedge tools and resources that are crucial for enhancing the scope and impact of our research. The DRAC supports researchers across various fields by providing access to high-performance computing, data storage, and specialized software platforms that facilitate large-scale data analysis, computational modelling, and collaborative research. In support of our initiative, our team has been honoured to be selected as part of the DRI Champions Program [?]. This includes a research grant for 2024–2025, from which a portion of the funding is specifically allocated to promoting the DRI, supporting research activities, and providing resources to facilitate this project's continued development and growth.

CONCLUSION

Our research paper demonstrated a DW utilization to efficiently and continuously prepare data for practical realtime data analysis and forecasting with ML algorithms, which we developed and tested in our previous research work [?], [16]–[20]. This paper discussed extraction, transformation, and loading process automation and the design and development of subsystems' integration for algorithmic trading ML modelling. Additionally, we discussed the design and implementation of the API, which is used by the core subsystem to access the DW.

Moreover, our research has emphasized that the architecture and design of an ETL system are pivotal in data warehousing. Our system exemplifies the importance of a well-structured and versatile design. Its current prototype

implementations have already demonstrated large data volumes, a variety of data types, and on-the-fly processing coupled with the stocks-price prediction systems that have already been demonstrated in our previous publications.

ACKNOWLEDGEMENTS

We thank Okanagan College and the OC's GIA Committee for funding and financial support of the applied student research projects. Additionally, we would like to thank the DRA of Canada for the DRI Champions Award.

Fig. 5. Data Fetching and Aggregation in Star Schema Design for ML Model Training

This work would not have been possible without the dedication and contributions of the students from Okanagan College, who were instrumental in developing various components, including ETL, database design, data collection, sanitization, and documentation. Key contributors include Jake Fischer, Jacob Rawlings, Alan Abdollahzadeh, Vanessa Dubouzet, Ben Carrier, Dominic Presch, Devon Volberg, Dylan Soares, Jaeden Soukoroff, Jacob Labelle, Parker Green, Dakota Flath, William (Noah) Blake, Yuan Hu, and Isaac Lengacher-Bergeron. Their efforts in earlier iterations and technical development, particularly by Nassi Ebadifard, Dakota Joiner, and Amy Vezeau, were vital to the success of this project. We also thank Dr. Vladimir Ryjov for providing technical insights and guidance throughout the research process for the previous version of prototype development in the Winter of 2024.

REFERENCES

A. A.-A. Falah Hassan, "Stock market index prediction using artificial neural network," Journal of Information Technology Research (JITR), vol. 15, pp. 1–16, 2022.

H. N. P. S. B. K. Geeta Kolte, Varadraj Kini, "Stock market prediction using deep learning," International Journal for Reseeaarch in Applied Science &

Engineering Technology, vol. 10, no. 4, pp. 26–32, 2022.

Y. Li, "Strategy analysis of financial neural network model in bond investment prediction," Frontiers in Business, Economics and Management, vol. 12, pp. 36–39, 2023.

R. K. Dubey, "Algorithmic Trading: The Intelligent Trading Systems and Its Impact on Trade Size," Expert Systems with Applications, vol. 202, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417422006479

S. Mart´ınez-Fernandez, J. Bogner, X. Franch, M. Oriol, J. Siebert,´ A. Trendowicz, A. M. Vollmer, and S. Wagner, "Software Engineering for AI-Based Systems: A Survey," ACM Transactions on Software Engineering and Methodology, vol. 31, no. 2, 4 2022.

T. Theate and D. Ernst, "An application of deep reinforcement´ learning to algorithmic trading," Expert Systems with Applications, vol. 173, p. 114632, 2021.

A. A. Yulianto, "Extract transform load (etl) process in distributed database academic data warehouse," APTIKOM Journal on Computer Science and Information Technologies, vol. 4, no. 2, pp. 61– 68, 2019.

O. Azeroual, G. Saake, and M. Abuosba, "Etl best practices for data quality checks in ris databases," in Informatics, vol. 6, no. 1. MDPI, 2019, p. 10.

W. A. A. C. C. O. O. C. O. . C. E. U. Adedoyin Tolulope Oyewole, Omotayo Bukola Adeoye, "Predicting stock market movements using neural networks: A review and application study," Computer Science & IT Research Journal, vol. 5, pp. 651–670, 2024.

E. M. Haryono, I. Gunawan, A. N. Hidayanto, U. Rahardja et al., "Comparison of the e-lt vs etl method in data warehouse implementation: A qualitative study,"

in 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS). IEEE, 2020, pp. 115–120.

A. Katari and A. Rodwal, "Next-generation etl in fintech: Leveraging ai and ml for intelligent data transformation," pp. 3491–3500, 2023.

M. Patel and D. B. Patel, "Progressive growth of etl tools: A literature review of past to equip future," Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS 2020, pp. 389–398, 2020.

N. Biswas, A. Sarkar, and K. C. Mondal, "Efficient incremental loading in etl processing for real-time data integration," Innovations in Systems and Software Engineering, vol. 16, pp. 53–61, 2020.

K. X. L. X. Xuekui Zhang, Yuying Huang, "Novel modelling strategies for high-frequency stock trading data," Financial Innovation, vol. 9, no. 39, pp. 1–25, 2023.

J. Wu, D. Bein, J. Huang, and S. Kurwadkar, "Etl and ml forecasting modeling process automation system," Applied Human Factors and Ergonomics International.

A. Wong, J. Figini, A. Raheem, G. Hains, Y. Khmelevsky, and P. C. Chu, "Forecasting of Stock Prices Using Machine Learning Models," in 2023 IEEE International Systems Conference (SysCon), 2023, pp. 1–7.

A. Wong, S. Whang, E. Sagre, N. Sachin, G. Dutra, Y.-W. Lim, G. Hains, Y. Khmelevsky, and F. Chang Zhang, "Short-Term Stock Price Forecasting using Exogenous Variables and Machine Learning Algorithms," in 2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA), 2023, pp. 260– 265.

A. Wong, J. Figini, A. Raheem, G. Hains, Y. Khmelevsky, and P. C. Chu, "Forecasting of stock prices using machine learning models," 2023 IEEE

International Ssytem Conferences (SYSCON), 2023.

A. Wong, S. Whang, E. Sagre, N. Sachin, G. Dutra, Y.-W. Lim, G. Hains, Y. Khmelevsky, and F. Zhang, "Short-term stock price forecasting using exogenous variables and machine learning algorithms," arXiv preprint arXiv:2309.00618, 2023.

A. Wong, J. Figini, A. Raheem, G. Hains, Y. Khmelevsky, and P. Chu, "Forecasting of Stock Prices Using Machine Learning Models," in IEEE International Systems Conference (SysCon) 2023, 2023.

1.	, Data	Incorrect punctuation	Correctness
2.	*´;´; "; " ; technologies'; company's; "; Canada's; ""; """; system's; FMP's; Fir's; CSV's; entry's; PostgreSQL's; timer's; DW's; model's; project's; subsystems'; OC's*	Text inconsistencies	Correctness
3.	*Abstract—This research focuses on the development of an automated data collection and transformation framework for Algorithmic Trading (AT) predictions using machine learning (ML).*	Paragraph can be improved	Clarity
4.	*The implementation of our computational resources is supported by the Digital Research Alliance of Canada (DRAC), Alliance Cloud Connect Pilot.*	Paragraph can be improved	Clarity
5.	*This*	Intricate text	Clarity
6.	~~largescale~~ → large-scale	Confused words	Correctness
7.	*real-time; realtime; Real-time*	Text inconsistencies	Correctness
8.	*Historical market data and real-time data streams are collected at 5minute intervals and stored in staging tables and will be aggregated into our model table as 15-min interval data.*	Ungrammatical sentence	Correctness
9.	*Historical market data and real-time data streams are collected at 5minute intervals and stored in staging tables and will be aggregated into our model table as 15-min interval data.*	Paragraph can be improved	Clarity
10.	~~important~~ → essential, vital, crucial	Word choice	Engagement

11.	~~remain~~ → remains	Faulty subject-verb agreement	Correctness
12.	*This paper discusses some of the solutions to acquiring complete historical and real-time data sets, along with the significant resources needed for storage and computation, in order to train a machine learning model for short-term Algorithmic Trading predictions.*	Paragraph can be improved	Clarity
13.	~~modeling~~ → modelling	Mixed dialects of English	Correctness
14.	*For our last phase, we will explore training and testing our machine learning model.*	Paragraph can be improved	Clarity
15.	~~coreyears~~ → core-years	Misspelled words	Correctness
16.	*It also allocates 16 VCPU-years, 4 Number of cloud instances, 32 GB of RAM, 7 volumes, 7 snapshots, 2 Floating IP addresses, 60,000 GB of cloud volume and snapshot storage of Cloud allocations on the Arbutuspersistent-cloud system.*	Ungrammatical sentence	Correctness
17.	*It also allocates 16 VCPU-years, 4 Number of cloud instances, 32 GB of RAM, 7 volumes, 7 snapshots, 2 Floating IP addresses, 60,000 GB of cloud volume and snapshot storage of Cloud allocations on the Arbutuspersistent-cloud system.*	Paragraph can be improved	Clarity
18.	, a	Incorrect punctuation	Correctness
19.	*After a project is approved and the resources are allocated a few more steps need to be taken to access these resources.*	Paragraph can be improved	Clarity
20.	~~important~~ → critical	Word choice	Engagement

21.	*Another important step was to set up Multifactor Authentication (MFA), which is used when we sign into DRAC and when you are using the SSH tunnel to access our server.*	Paragraph can be improved	Clarity
22.	~~certain~~ → specific	Word choice	Engagement
23.	*Furthermore, users will have to send an email to DRAC, requesting access to certain resources and specifying the quantity required, and wait for a reply before they can create and connect to their PostgreSQL database.*	Paragraph can be improved	Clarity
24.	, howto	Incorrect punctuation	Correctness
25.	~~howto~~ → how to	Confused words	Correctness
26.	, and	Incorrect punctuation	Correctness
27.	*Our previous work only included the three previous years.*	Paragraph can be improved	Clarity
28.	*Our new data set will be ten times larger than our previous set.*	Paragraph can be improved	Clarity
29.	, and	Incorrect punctuation	Correctness
30.	*One of our biggest challenges is related to the fact that many of the previous sets of data are incomplete and they are leading to unexpected results with the machine learning model.*	Paragraph can be improved	Clarity
31.	lack of development of	Paragraph can be improved	Clarity
32.	~~The main~~ → the main	Confused words	Correctness

33.	*Many algorithmic trading and ML forecasting projects were limited by the size of the data set in model development and by the lack of system support in their possible implementation of the developed models.*	Paragraph can be improved	Clarity
34.	]-	Incorrect punctuation	Correctness
35.	*For ML model development, the required data for training and testing are collected from different sources.*	Paragraph can be improved	Clarity
36.	*Capturing heterogeneous data from different sources and storing it in single or multiple data depositories is a common issue for many organizations.*	Ungrammatical sentence	Correctness
37.	*Capturing heterogeneous data from different sources and storing it in single or multiple data depositories is a common issue for many organizations.*	Paragraph can be improved	Clarity
38.	*along with the volume and speed at which financial data is generated*	Misplaced words or phrases	Correctness
39.	*The paper states that market complexity and volatility, along with the volume and speed at which financial data is generated, shape the challenges surrounding stock market analysis.*	Paragraph can be improved	Clarity
40.	*These challenges underscore the need for a system that can handle large datasets in a way that supports accurate realtime predictions in a high-frequency trading environment.*	Ungrammatical sentence	Correctness

41.	*These challenges underscore the need for a system that can handle large datasets in a way that supports accurate realtime predictions in a high-frequency trading environment.*	Paragraph can be improved	Clarity
42.	al.,	Incorrect punctuation	Correctness
43.	These are	Pronoun use	Correctness
44.	*Furthermore, Katari and Rodwal [11] in their paper discuss the inadequacy of traditional ETL processes in handling the volume and inconsistency of the data inherent to financial markets.*	Paragraph can be improved	Clarity
45.	~~had~~ → has	Incorrect verb forms	Correctness
46.	*They state that harnessing the capability of AI and ML techniques had advantages relating to enhanced accuracy, scalability, speed, and integration, among other benefits.*	Paragraph can be improved	Clarity
47.	~~them~~ → which	Pronoun use	Correctness
48.	al.,	Incorrect punctuation	Correctness
49.	*According to Biswas et al., [13], many organizations use commercially available, GUI-based commercial products as the ETL solution.*	Paragraph can be improved	Clarity
50.	~~Cloudbased~~ → Cloud-based	Misspelled words	Correctness
51.	dataset.	Closing punctuation	Correctness
52.	*model; Model; model's*	Text inconsistencies	Correctness
53.	improved data-loading efficiency	Paragraph can be improved	Clarity

54.	*NGOC-BAO-VAN et al. [?] discussed designing and implementing a coffee commodity trading DW to support informed decision-making.*	Paragraph can be improved	Clarity
55.	*After data extraction, the ETL process extracts, cleans, and transforms the data from multiple sources.*	Paragraph can be improved	Clarity
56.	Price trend visualization and analysis	Paragraph can be improved	Clarity
57.	, but	Incorrect punctuation	Correctness
58.	~~objectoriented~~ → object-oriented	Misspelled words	Correctness
59.	*In order to have accurate short term forecasting, the model needs to always have knowledge of recent market activity.*	Paragraph can be improved	Clarity
60.	~~short term~~ → short-term	Confused words	Correctness
61.	~~have knowledge of~~ → know	Wordy sentences	Clarity
62.	always to have knowledge of recent market activity	Inappropriate colloquialisms	Delivery
63.	~~gb~~ → GB	Confused words	Correctness
64.	*cloud; Cloud*	Text inconsistencies	Correctness
65.	~~needed~~	Incorrect verb forms	Correctness
66.	*This*	Intricate text	Clarity
67.	missed,	Incorrect punctuation	Correctness
68.	performance history	Paragraph can be improved	Clarity

69.	scheduling,	Incorrect punctuation	Correctness
70.	*Timer; timer; timer's*	Text inconsistencies	Correctness
71.	*service; Service*	Text inconsistencies	Correctness
72.	*each execution, a time window is created which covers the most recent trading period, with a small overlap from the previous run to ensure no data are missed.*	Ungrammatical sentence	Correctness
73.	~~small~~ → slight	Word choice	Engagement
74.	missed.	Closing punctuation	Correctness
75.	~~1 hour~~ → 1-hour	Confused words	Correctness
76.	~~database~~ → Database	Improper formatting	Correctness
77.	*After retrieving recent data from FMP, the results are written to the database with an "upsert" operation, where entries are updated when already present, and inserted when not present yet.*	Paragraph can be improved	Clarity
78.	*Listing 4 shows the portion of the collector that filters each record into the active time window and writes it to the corresponding ticker table using an UPSERT operation.*	Paragraph can be improved	Clarity
79.	~~5 minute~~ → 5-minute	Confused words	Correctness
80.	~~record~~ → Record	Improper formatting	Correctness
81.	and return	Conjunction use	Correctness
82.	, and	Incorrect punctuation	Correctness

83.	*Data is extracted from the FMP API using this time window and the schedule is timed so that each collection happens shortly after new data is available.*	Paragraph can be improved	Clarity
84.	*This*	Intricate text	Clarity
85.	, and	Incorrect punctuation	Correctness
86.	*The script uses lists of tickers so that symbols we want to track could be added or removed at any time and the collector would adjust.*	Paragraph can be improved	Clarity
87.	*This*	Intricate text	Clarity
88.	*table; Table*	Text inconsistencies	Correctness
89.	~~cloud based~~ → cloud-based	Confused words	Correctness
90.	~~to always be~~ → always to be	Inappropriate colloquialisms	Delivery
91.	*Running this process with systemd timers on the cloud based VM ensures a stable environment for collection, allowing the historical dataset to always be up to date, accurate, and ready for training the forecasting model.*	Paragraph can be improved	Clarity
92.	*Each table uses OHLCV format (open, high, low, close, and volume), matching the structure provided by our data source (FMP API).*	Paragraph can be improved	Clarity
93.	~~5 minute~~ → 5-minute	Confused words	Correctness

94.	*Listing 5 shows the structure of a typical staging table, where the timestamp acts as the primary key and ensures that overlapping collection windows do not introduce duplicates.*	Paragraph can be improved	Clarity
95.	. aapl	Improper formatting	Correctness
96.	to update rows when collection windows overlap easily	Inappropriate colloquialisms	Delivery
97.	~~were to be~~ → are	Paragraph can be improved	Clarity
98.	~~auto-collected~~ → auto-collected	Confused words	Correctness
99.	*In a later step of the pipeline, both the historical data and the auto collected staging data are combined into a single model table used for training the forecasting model.*	Paragraph can be improved	Clarity
100.	~~important~~ → essential	Word choice	Engagement
101.	*Testing and logging are important parts of making sure the auto collector captures 100% of the expected market data for each market day.*	Paragraph can be improved	Clarity
102.	*Its job is to check that every 5 minute interval during standard trading hours was collected for every ticker.*	Ungrammatical sentence	Correctness
103.	*Its job is to check that every 5 minute interval during standard trading hours was collected for every ticker.*	Paragraph can be improved	Clarity
104.	~~five-minute~~ → five-minute	Confused words	Correctness
105.	day,	Incorrect punctuation	Correctness

106.	. execute	Improper formatting	Correctness
107.	*The test knows the start date of collection and adjusts its expectations as the days go on, so it continuously logs a coverage percentage for each ticker.*	Paragraph can be improved	Clarity
108.	*This*	Intricate text	Clarity
109.	~~of DRI~~ → the DRI	Incorrect phrasing	Correctness
110.	the database	Determiner use (a/an/the/this, etc.)	Correctness
111.	, testing	Incorrect punctuation	Correctness
112.	, and	Incorrect punctuation	Correctness
113.	the ELT	Determiner use (a/an/the/this, etc.)	Correctness
114.	*Follow ELT Process, instead of ETL.*	Paragraph can be improved	Clarity
115.	~~. Used~~ → —used	Incomplete sentences	Delivery
116.	the automatic	Determiner use (a/an/the/this, etc.)	Correctness
117.	the model	Determiner use (a/an/the/this, etc.)	Correctness
118.	the Table	Determiner use (a/an/the/this, etc.)	Correctness
119.	The User	Determiner use (a/an/the/this, etc.)	Correctness

120.	*From From Staging Tables to Model Table in Detail: A Jupyter Notebook will extract the raw data from the staging tables, merge and transform the data and load them into a single table for model training.*	Paragraph can be improved	Clarity
121.	~~From~~ From	Wrong or missing prepositions	Correctness
122.	*Data Formatting: The extracted data must be formatted to align with the DW schema, including proper conversion of date formats, numerical precision, and standardization of measurement units (e.g., currency, percentages).*	Paragraph can be improved	Clarity
123.	*Numerical values are normalized during the model training process.*	Paragraph can be improved	Clarity
124.	~~disabled~~ → turned off	Potentially sensitive language	Delivery
125.	, and	Incorrect punctuation	Correctness
126.	*Fir's crontab tool has been disabled and Fir has minimum requirements on job length.*	Paragraph can be improved	Clarity
127.	~~is~~ → are	Faulty subject-verb agreement	Correctness
128.	*The disjoint databases and short runtime lengths of many of our scripts require some of the work to be done on Arbutus Cloud and on local (personal) machines, and thus some form of synchronization is required.*	Paragraph can be improved	Clarity
129.	*The FMP API, configured to retrieve stock information across 15 minute intraday intervals, will return a maximum of 10 days of data for one stock symbol, per API call.*	Ungrammatical sentence	Correctness

130.	*The FMP API, configured to retrieve stock information across 15 minute intraday intervals, will return a maximum of 10 days of data for one stock symbol, per API call.*	Paragraph can be improved	Clarity
131.	an end	Determiner use (a/an/the/this, etc.)	Correctness
132.	*We begin by defining a start date, end date, and stock symbols.*	Paragraph can be improved	Clarity
133.	, calling	Incorrect punctuation	Correctness
134.	*Then, for each stock symbol, we have an inner loop which iterates through the target range calling the API with distinct 5-day windows.*	Paragraph can be improved	Clarity
135.	*The resulting CSV files become input for a script which checks for missing entries.*	Paragraph can be improved	Clarity
136.	~~to not~~ → not to	Misplaced words or phrases	Correctness
137.	not to trigger false alerts	Inappropriate colloquialisms	Delivery
138.	*Exclusion dates are inserted into a list (holidays, etc.) so as to not trigger false alerts.*	Paragraph can be improved	Clarity
139.	*Expected 5-minute intervals are calculated, then actual results are then compared against the generated list.*	Ungrammatical sentence	Correctness
140.	*Expected 5-minute intervals are calculated, then actual results are then compared against the generated list.*	Paragraph can be improved	Clarity

141.	*Occasionally, FMP does not have data for an interval but it is critical to validate the missing data is due to FMP's data collection process and not a connection issue.*	Ungrammatical sentence	Correctness
142.	*The raw data CSV's are now ready to be patched (missing entries filled with previous entry's information) and/or uploaded to the database.*	Ungrammatical sentence	Correctness
143.	*The CSV stock data can now be inserted into our Fir hosted database using automated python scripts.*	Ungrammatical sentence	Correctness
144.	*The CSV stock data can now be inserted into our Fir hosted database using automated python scripts.*	Paragraph can be improved	Clarity
145.	database,	Incorrect punctuation	Correctness
146.	*Before attempting to insert data into the database an SSH Tunnel needs to be made to Fir so the database is accessible to us.*	Paragraph can be improved	Clarity
147.	*As using multiple INSERT commands can take over 10 minutes per stock due to overhead, we take advantage of PostgreSQL's COPY command which acts as a bulk insert and takes approximately 5 seconds per stock.*	Paragraph can be improved	Clarity
148.	, which	Incorrect punctuation	Correctness
149.	~~throught~~ → through	Misspelled words	Correctness
150.	*In order to update the database with the latest stock information, the FMP API must be called continually throughout the day.*	Paragraph can be improved	Clarity

151.	*The virtual machine on Arbutus Cloud, running a Bash shell on AlmaLinux, provides the mechanism through which our Python script is automated.*	Paragraph can be improved	Clarity
152.	Section.	Closing punctuation	Correctness
153.	*We are then able to open a secure SSH tunnel to the Fir database, using a non-dedicated port on the Arbutus VM as our local port.*	Paragraph can be improved	Clarity
154.	*Finally, once the port-forwarding is setup is complete, we can use the Arbutus terminal to run psql using the now open ssh tunnel.*	Ungrammatical sentence	Correctness
155.	*Finally, once the port-forwarding is setup is complete, we can use the Arbutus terminal to run psql using the now open ssh tunnel.*	Paragraph can be improved	Clarity
156.	*This*	Intricate text	Clarity
157.	*This completes the connection and we have command line access to the PostgreSQL database on Fir.*	Ungrammatical sentence	Correctness
158.	*This completes the connection and we have command line access to the PostgreSQL database on Fir.*	Paragraph can be improved	Clarity
159.	two-factor authentication	Confused words	Correctness
160.	~~include~~ → includes	Faulty subject-verb agreement	Correctness
161.		Tone suggestions	Delivery
162.	, which	Incorrect punctuation	Correctness

163.	*Preferably, we would like to connect Arbutus to Fir exclusively through internal DRAC tools or networks, as opposed to SHH which always requires Duo authentication.*	Paragraph can be improved	Clarity
164.	, which	Incorrect punctuation	Correctness
165.	, as	Incorrect punctuation	Correctness
166.	*This table can only consist of numerical data as machine learning models are highly mathematical in nature.*	Paragraph can be improved	Clarity
167.	~~python~~ → Python	Confused words	Correctness
168.	*Although processing could be done on the Fir database using PL/pgSQL, it would be much harder to accomplish without the extensive python libraries that make data transformation easy.*	Paragraph can be improved	Clarity
169.	~~stocks~~ → stock's	Incorrect noun number	Correctness
170.	*Each stocks data is read in from the table and can now be joined to the sector data using pandas.*	Paragraph can be improved	Clarity
171.	~~individual~~	Wordy sentences	Clarity
172.	, so	Incorrect punctuation	Correctness
173.	can track	Incorrect verb forms	Correctness
174.	~~pre~~ → pre-	Misspelled words	Correctness
175.	~~post-holidays~~ → post-holidays	Confused words	Correctness
176.	holidays.	Closing punctuation	Correctness

177.	first,	Incorrect punctuation	Correctness
178.	the database	Determiner use (a/an/the/this, etc.)	Correctness
179.	*Currently, the DW only contains data for 2021 to 2024, but the FMP offers a historical dataset spanning 30 years.*	Paragraph can be improved	Clarity
180.	*To estimate these requirements, the total number of records in the fact table for forty years of data must be calculated, along with the average size of a single record in the fact table.*	Paragraph can be improved	Clarity
181.	~~is~~ → are	Faulty subject-verb agreement	Correctness
182.	*Along with the DW connection, API routes created using Flask provide access to required data types through their respective routes, such as stocks, financial indexes, and commodities.*	Paragraph can be improved	Clarity
183.	*The core subsystem uses this API to query data from the DW based on controls provided on the web UI to filter data based on requirements.*	Paragraph can be improved	Clarity
184.	, as	Incorrect punctuation	Correctness
185.	*The queried data is then presented on the web UI with the help of graphs as shown in Fig. 4.*	Paragraph can be improved	Clarity

186.	*Fig. 4 shows the comparison between the red line, which represents the actual stock prices, and the blue line, indicating the predicted prices over time, provides an analysis of the model's accuracy, where periods of close alignment between the lines suggest successful prediction, while deviations ...*	Ungrammatical sentence	Correctness
187.	*Fig. 4 shows the comparison between the red line, which represents the actual stock prices, and the blue line, indicating the predicted prices over time, provides an analysis of the model's accuracy, where periods of close alignment between the lines suggest successful prediction, while deviations ...*	Paragraph can be improved	Clarity
188.	*This visualization highlights the model's effectiveness in capturing general stock price movement trends while underscoring challenges in predicting specific fluctuations accurately.*	Paragraph can be improved	Clarity
189.	*The underlying hardware infrastructure significantly influences the performance of each subsystem.*	Paragraph can be improved	Clarity
190.	*However, as the system is migrated to the Cloud, we will be able to scale both the volume of data processed and the frequency of predictions, leading to further optimizations and improvements in performance.*	Paragraph can be improved	Clarity
191.	*We re-designed the DW, implementing a star schema for efficient data querying.*	Paragraph can be improved	Clarity

192.	*This schema is tailored to the needs of our ML algorithm, ensuring that the data is preaggregated and ready for use.*	Paragraph can be improved	Clarity
193.	~~simpler~~ → more straightforward	Word choice	Engagement
194.	*This design reduces the need to join multiple data from tables and improves the overall efficiency of the ETL process, leading to a simpler API to provide data access for ML model training.*	Paragraph can be improved	Clarity
195.	*The updated workflow can be seen in Fig. 5, modelled after the work from [?].*	Paragraph can be improved	Clarity
196.	of models	Wrong or missing prepositions	Correctness
197.	*In future research, we will use 30 years of financial data for the training and testing models.*	Paragraph can be improved	Clarity
198.	~~cuttingedge~~ → cutting-edge	Misspelled words	Correctness
199.	~~as part of~~ → for	Paragraph can be improved	Clarity
200.	*This*	Intricate text	Clarity
201.	*Our research paper demonstrated a DW utilization to efficiently and continuously prepare data for practical realtime data analysis and forecasting with ML algorithms, which we developed and tested in our previous research work [?], [16]–[20].*	Ungrammatical sentence	Correctness

202.	*Our research paper demonstrated a DW utilization to efficiently and continuously prepare data for practical realtime data analysis and forecasting with ML algorithms, which we developed and tested in our previous research work [?], [16]–[20].*	Paragraph can be improved	Clarity
203.	the extraction	Determiner use (a/an/the/this, etc.)	Correctness
204.	*This paper discussed extraction, transformation, and loading process automation and the design and development of subsystems' integration for algorithmic trading ML modelling.*	Paragraph can be improved	Clarity
205.	~~, which is~~	Paragraph can be improved	Clarity
206.	~~stocks-price~~ → stock price	Confused words	Correctness
207.	*Its current prototype implementations have already demonstrated large data volumes, a variety of data types, and on-the-fly processing coupled with the stocks-price prediction systems that have already been demonstrated in our previous publications.*	Paragraph can be improved	Clarity
208.	*This work would not have been possible without the dedication and contributions of the students from Okanagan College, who were instrumental in developing various components, including ETL, database design, data collection, sanitization, and documentation.*	Paragraph can be improved	Clarity

209.	*Their efforts in earlier iterations and technical development, particularly by Nassi Ebadifard, Dakota Joiner, and Amy Vezeau, were vital to the success of this project.*	Paragraph can be improved	Clarity
210.	~~Reseeaarch~~ → Research	Misspelled words	Correctness
211.	,-	Incorrect punctuation	Correctness
212.	reinforcement-	Incorrect punctuation	Correctness
213.	~~etl~~ → ETL	Confused words	Correctness
214.	*O. Azeroual, G. Saake, and M. Abuosba, "Etl best practices for data quality checks in ris databases," in Informatics, vol. 6, no. 1. MDPI, 2019, p. 10.*	Ungrammatical sentence	Correctness
215.	*A. Katari and A. Rodwal, "Next-generation etl in fintech: Leveraging ai and ml for intelligent data transformation," pp. 3491–3500, 2023.*	Ungrammatical sentence	Correctness
216.	~~etl~~ → ETL	Confused words	Correctness
217.	~~etl~~ → ETL	Confused words	Correctness
218.	*J. Wu, D. Bein, J. Huang, and S. Kurwadkar, "Etl and ml forecasting modeling process automation system," Applied Human Factors and Ergonomics International.*	Ungrammatical sentence	Correctness
219.	~~Ssytem~~ → System	Misspelled words	Correctness

220.	*The main contributions of this paper are (1) a*	Efficient Discontinuous Phrase-Structure Parsing via the Generalized Maximum Spanning Arborescence - ACL Anthology https://aclanthology.org/D17-1172/	Originality
221.	*a vital role in determining the implementation of*	Community Development | Gabriel Project Mumbai https://www.gabrielprojectmumbai.org/community-development	Originality
222.	*iques. Wu et al. [15] propose a system using Apache Airflow in combination with Python scripts to orchestrate and automate ETL tasks. Their framework demonstrates the advantages of scheduling ETL processes at regular intervals, automatically extracting data from financial sources, transforming it t...*		Originality
223.	*om DW. While this model operates on a smaller scale and updates less frequently than high-frequency trading systems require, it highlights the benefits of automation for consistency, error reduction, and efficiency in data loading. Our previous analyses on several implementation techniques were doc...*		Originality
224.	*(DRI). The system runs on a p4-6gb instance, which provides 4 virtual CPUs, 6 GB of RAM, and a 20 GB persistent disk. Hosting the process in the cloud allows for consistent uptime, simple remote access, and protection against local hardware or power failures. Arbutus Cloud instances offer persisten...*		Originality

225.	*data. Job scheduling is managed using systemd timers, which run data collection at fixed intervals during standard market hours. This scheduling method ensures no data is missed, and reliable execution in the event of a network or system interruption. Each run creates a status log, forming a histo...*	Originality
226.	*target The timer triggers the service shown in Listing 2, which executes the collector script. Listing 2. Systemd service that runs the Auto Data Collector [Unit] Description=AutoDataCollector Service # Ensure network is ready before running After=network.target [Service] # Run the script once per ...*	Originality
227.	*djust. This makes the system flexible and easier to maintain as data requirements evolve. Because FMP provides timestamps in EST, the script preserves this format when inserting records into the PostgreSQL rows.append((ts, record["open"], record["high"], record["low"], record["close"], record["vol...*	Originality
228.	*stamp. Listing 5 shows the structure of a typical staging table, where the timestamp acts as the primary key and ensures that overlapping collection windows do not introduce duplicates. Listing 5. Example staging table used for a single ticker -- Staging table for a single ticker (AAPL) CREATE TABL...*	Originality

229.	*al = 0 table = format_table_name(symbol) current = start_date with conn.cursor() as cur: while current <= end_date: # Skip weekends if current.weekday() >= 5: current += timedelta(days=1) continue # Intervals for one trading day total_expected += expected_per_day # Count found rows for that day ope...*	Originality
230.	*e_dt = datetime.combine(current, time(16,0)).replace(tzinfo=None) cur.execute(f"SELECT COUNT(*) FROM {table} WHERE ts >= %s AND ts < %s", (open_dt, close_dt),) total_actual += cur.fetchone()[0] current += timedelta(days=1) # Return coverage percentage return (total_actual / total_expected) * 100 ...*	Originality
231.	*else 0 The test knows the start date of collection and adjusts its expectations as the days go on, so it continuously logs a coverage percentage for each ticker. This helped us catch several issues early, such as missing intervals, FMP API changes, and unexpected execution times. Finding these prob...*	Originality
232.	*clean. In addition to the coverage test, the collector writes a status log for every scheduled run, noting when the job started, when it finished, how long it took and whether any errors occurred. The coverage test also records its results in the logs. Together, these tests and logs provide essenti...*	Originality

233.	*e code Fir's crontab tool has been disabled and Fir has minimum requirements on job length. The virtual machine on Arbutus Cloud provides access to scheduled tasks, which is necessary to run our scripts at regular inte*		Originality
234.	*00 pm Eastern Standard Time, Monday through Friday.*	Multiple Formulation Related Laboratory Tools And Equipment (brand Name Or Equal) [Tender documents : T488000434]	Originality
235.	*dataset. Estimating the DW's storage requirements for this expansion is critical to identifying storage capacity, query response times, and cost management. Furthermore, protecting the space needed to accommodate ten years of data records is es*		Originality
236.	*eporting Fig. 4 shows the comparison between the red line, which represents the actual stock prices, and the blue line, indicating the predicted prices over time, provides an analysis of the model's accuracy, where periods of close alignment between the lines suggest successful prediction, while de...*		Originality
237.	*ovement. This visualization highlights the model's effectiveness in capturing general stock price movement trends while underscoring challenges in predicting specific fluctuations accurately. Fig. 4. An Example of DRI DBMS and ML Integration: Forecast Report for Analysis of Stock Prices of Apple, ...*		Originality

238.	*Microsoft The underlying hardware infrastructure significantly influences the performance of each subsystem. For example, storage drives with high Input/Output Operations Per Second (IOPS) can drastically improve the access time to the DW [?]. In contrast, GPUs with higher CUDA cores accelerate ML ...*		Originality
239.	*asks [?]. In the project's current phase, we are utilizing the hardware resources available within our Computer Science Department, which are sufficient for our limited-scale deployment. However, as the system is migrated to the Cloud, we will be able to scale both the volume of data processed and ...*		Originality
240.	*providing access to high-performance computing, data storage, and*	Economic Times Vision Conclave: Developing full ecosystem to leverage the emerging AI wave, ETGovernment https://government.economictimes.indiatimes.com/news/digital-india/economic-times-vision-conclave-developing-full-ecosystem-to-leverage-the-emerging-ai-wave/105000947	Originality
241.	*Training This work would not have been possible without the dedication and contributions of the students from Okanagan College, who were instrumental in developing various components, including ETL, database design, data collection, sanitization, and docum*		Originality
242.	*J. Bogner, X. Franch, M. Oriol, J. Siebert,´ A. Trendowicz, A. M. Vollmer,*	DOGO4ML: Development, operation and data governance for ML-based software systems	Originality

243.	*S. Wagner, "Software Engineering for AI-Based Systems: A Survey," ACM Transactions on Software Engineering and Methodology, vol. 31, no. 2,*	DOGO4ML: Development, operation and data governance for ML-based software systems	Originality
244.	*An application of deep reinforcement ´ learning to algorithmic trading," Expert Systems with Applications,*	Using Deep Reinforcement Learning with Hierarchical Risk Parity for Portfolio Optimization	Originality
245.	*A. A. Yulianto, "Extract transform load (etl) process in distributed database academic data warehouse," APTIKOM Journal on Computer Science and Information Technologies, vol. 4, no. 2, pp.*	Tiny datablock in saving Hadoop distributed file system wasted memory	Originality
246.	*O. Azeroual, G. Saake, and M. Abuosba, "Etl best practices for data quality checks in ris databases,*	Solving problems of research information heterogeneity during integration – using the European CERIF and German RCD standards as examples	Originality
247.	*E. M. Haryono, I. Gunawan, A. N. Hidayanto,*	Quality Analysis Of Digital Business Services In Improving Customer Satisfaction | Startupreneur Business Digital (SABDA Journal) https://journal.pandawan.id/sabda/article/view/119	Originality
248.	*Comparison of the e-lt vs etl method in data warehouse implementation: A qualitative study," in 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS*	Quality Analysis Of Digital Business Services In Improving Customer Satisfaction | Startupreneur Business Digital (SABDA Journal) https://journal.pandawan.id/sabda/article/view/119	Originality
249.	*Progressive growth of etl tools: A literature review of past to equip future," Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS 2020, pp. 389–398,*	Airflow Dag Automation in Distributed Etl Environments	Originality

250.	*N. Biswas, A. Sarkar, and K. C. Mondal, "Efficient incremental loading in etl processing for real-time data integration," Innovations in Systems and Software Engineering, vol. 16, pp. 53–61, 2020.*	OLAP Mining with Educational Data Mart to Predict Students' Performance | Najm | Informatica https://informatica.si/index.php/informatica/article/view/3853/0	Originality