Proceedings of the International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS 2023)
DVD Part Number: CFP22DN7-DVD; ISBN: 979-8-3503-0090-1

A Model for the Analytical Performance of Data Lake in Stock Market Analysis with Databricks Delta Lake

Dr. S. Kamalakkannan, Associate Professor
Department of Information Technology
School of Computing Sciences,
Vels Institute of Science, Technology & Advanced Studies
Pallavaram, Chennai, India
kannan.scs@velsuniv.ac.in

A.Yasmin, Assistant Professor
Department of Computer Science ,
JBAS College for women
Chennai, India
yasumiza@gmail.com

Dr.Arunkumar.R, Lecturer- Digital Forensics & Cyber Security
Faculty of Computing, Engineering and Science,
University of South Wales,
Treforest/Newport Campus, United Kingdom
arun.kumar@southwales.ac.uk

Dr. P. Kavitha, Assistant Professor
Department of Computer Applications
School of Computing Sciences,
Vels Institute of Science, Technology & Advanced Studies
Pallavaram, Chennai, India
pkavikamal@gmail.com

Abstract— **Stock market investments are highly rewarding but also high in risk. Modern investors use variety of tools to take informed investment decisions. In the current era of digital world, financial service industry has generated huge volume and immense verities of data with extreme speed. Due to the rapid growth in data collection and the heterogeneous nature and complexity of the data, there is a need for Big Data analytical solution that would be able to deal with the stock market data. Large volumes of unstructured, heterogeneous raw data can be stored in a massively scalable manner using data lakes, which are the ideal solution to the big data storage conundrum. The ability of a data lake to preserve data in its original format while processing it at runtime using a schema on-read technique is its key feature. The challenge faced in the data lake is performing analytics which is a significant tool to calculate and analyze the stock market. The proposed architecture of Azure Databricks DeltaLake (ADDL) with Azure DataLake Storage Generation 2 (ADLSG2) is used for analytical processes like Fibonacci retracement for better stock analysis, which aid in forecasting the market price for better investment. As a result, the research focus is to produce a storage having read as well as write capabilities by taking into consideration the Extract-Load-Transform (ELT) operation on the datasource. In this experimental databricks implementation, runtime is performed using open source of Apache Spark API and a highly improved execution engine, which results in a significant performance improvement when comparing to the standard source of Apache Spark available on the ADLS platform. Additionally, the Fibonacci retracement level calculation is achieved with the analytics and forecasting of test close price with various ML and DL techniques such as KNN, LSTM are compared with original price of the test data for better prediction of forecast close price.**

Keywords: Big data analytics, ADLSG2, Databricks Delta Lake, ELT, stock market, Fibonacci retracement, LSTM, Apache spark

I. INTRODUCTION

Investing in stocks involves taking risk. Modern investors rely on a variety of tools to make judgments. Apart from fundamental analysis, technical analysis that employs several tools for a detailed assessment of developments in Stock market and predicts the future trends and fluctuations in price is becoming increasingly popular. A trend is the general pattern of the price of a market or an asset. Trend does not move in a straight line. We can understand the market trend by looking at longer-term pricing trends. A stock is said to be in an 'Uptrend' when the direction of the price movement is upwards. An easy way to identify an 'Uptrend' is to see if the stock is going above its previous high and not falling below its previous low. A stock is said to be in a 'Downtrend' when the direction of the price movement is downwards. An easy way to identify a stock in 'Downtrend' is to see if the stock goes down, rises, and then goes below its previous low. The finance services business has amassed vast amounts of data at rapid speed, creating the fresh crop of data analytical instances known as Big Data Analytics (BDA). Customer demand for data computing as well as increase in growth of data volume ultimately result in the emergence of big data concept [1, 2]. The evolution of the big data concept may be influenced by data volume [1, 2]. Big data is defined by six characteristics: volume, variability, velocity, veracity, value, and variety [3]. A data lake is also intended to enable the capability to reliably do analytics, batch processing, and real-time analysis on enormous volume of data. [4][5][6]. This is accomplished by uniting the advantages of SQL and NoSQL database approaches and supplementing them with OLTP and OLAP capabilities. All the data components of data lake are identified by an unique ID as well as frequently include more metadata.

Data lakes help organizations improve raw data archival, capture, exploration, and refinement. A data lake serves as a scalable solution for data storing and analyzing retained data in its native form, which is then utilized for extracting meaningful information [7]. As a result, data lakes cannot be built from scratch or established employing conventional software applications [8]. As a result, numerous data lake solutions have been using Apache Hadoop as a core platform. [9].

Delta Lake was created in 2016 by Databricks to aid in the storage of data with the relevant items are regarded as a portion of a table over Data Lake is executed as a Parquet

transaction log. This allows data to load to billions of components per table [10]. Delta Lake supports as data layer and is an open source project that provides reliability to Data Lake. This technology has created solid data lakes as well as an essential component in the development of a cloud data environment. Delta Lake merged streaming as well as batch data processing that can provide Atomicity Consistency Isolation Durability (ACID) transactions and handle scalable metadata. Delta Lake is built over the established data lake as well as complies with Apache Spark APIs. Delta lakes can be configured on Databricks based on workload patterns. Adding a transaction log that starts with an entry that describes Every single one of the current files. Delta Lake is capable of transforming a current directory of Parquet files into the Delta Lake table with zero copies. Delta Lake made significant progress in development during 2019 as by simply completing 50% computed hours on Databricks. The metadata layer is naturally setup to adopt the enforcement features of data quality. Delta Lakes process information in a very distinct sequence, while it continues to keep the data in its original form. The preprocessing procedure does not end until the data is demanded from the application or when the query is run. Consequently, it contradicts the data warehouse's standard Extract Transform Load (ETL) principles while promoting the newer ELT data handling order. There isn't any established data structure in Delta Lake, and data is extracted from sources and transferred to data lakes according to the absolutely necessary parameters . As a result, Delta Lake can deal with both of those write-intensive as well as read-intensive demands , such as transaction and analytic, combining the two divergent considerations such as Reading as well as writing. A key application of Delta Lake is the schema enforcement function that controls that the data that is uploaded in a database conforms to the schema of the lake as well as constraints API. This enables that the data table owners can set limit levels on the amount of information that can be accumulated. Entries which exceed the limits could programmatically be rejected or even quarantined by the client libraries of Delta lake. Such basic characteristics are proving to be highly efficient in enhancing the quality of pipelines based on Data Lakes. The present study concentrates about Azure Databricks Delta Lake (ADDL) to efficiently execute BDA without data loss from Azure Data Lake Storage Gen2 (ADLSG2) and transform to the same ADLSG2 as user end based on analytical processes such as Fibonacci retracement for better stock analysis. ADDL is implicated in ADLSG2 since it is irreversible and does not run analyticals; however, ADDL assists in performing BDA processes and delivering them to the user.

A. Challenges of the Study

Rapid human civilization advancement, combined with leading technological achievements, has significantly established finance support service businesses like insurance, banking, Shares trading, mutual fund investments, etc. Globalization has increased the number of desires and requirements. Financial service industries require better stock price prediction tools to increase profits and avoid losses in the stock market. Large data analyzed to provide improved user experience as well as data handling. An efficient platform is required to generate immense verities of information at rapid pace that lead to the development of a modern era of data processing instances called Big Data Analytics.

B. Problem Statement

The stock market has traditionally been regarded as a statistical puzzle. People believe that knowing a market's statistics allows us gain from investment. An essential factor to follow before investment is to perform research that assist in assuming the value of future investment. Thus, the digital footprints have pitched for an analysis using the analytics that assist business judgments that are thorough and justified . The client demand for data processing, combined with the rapid surge in the volume of data, resulted in the evolution of the big data concept. Data Lakes are indeed a advantageous answer to the big data storage conundrum, serving as a highly expandable storage achieve with capabilities to store immense quantities of raw data as in un - structured and diverse manner. The capability of a data lake to store data in native form and process the data at runtime using a schema on-read technique is advantageous. The challenge faced in the data lake is performing analytics which is a significant tool to calculate and analyze the stock market for better prediction of stock prices. So, the research focus on Delta Lake platform for analytical process like Fibonacci retracement for analysis and forecasting the market price.

C. Objective of the Research work

For reliable prediction and analysis of large data, BDA can be applied in a variety of disciplines. They make it easier to find important information in huge volume of data that would otherwise be obscured. This study focuses on the solution of a stock market analysis to comprehend its volatile character and forecast its behavior to earn by investing in it.

1) To integrate both batch data and streaming process, Delta Lake is performed to provide ACID transaction and maintain scalable metadata.
2) To progress analytical process in time series data of stock market, the forecasting method is initiated through Machine Learning (ML) and Deep Learning (DL) techniques for better prediction of forecasted close price.
3) To determine the set price target or stop loss level for trading, Fibonacci retracement is calculated for all available stock price before investment.

To produce a storage with read as well as write capabilities based on ELT operations on the data source, the proposed architecture of ADDL with ADLSG2 is suggested in stock market analysis.

D. Scope of Work

This research scope is to focus on obtaining better analysis in the stock market and help investors to do better investment in it. The study of predicting future support as well as resistance levels according to historical price patterns as well as reversals is defined as Fibonacci analysis. When a currency pair changes trend, forex traders are incredibly inquisitive about how much the pair is going to move in the new direction. When determining how far a currency pair will retrace, or move against, a prior trend, certain Fibonacci ratios might be useful. The ML and DL technique assist to understand the time series dataset by data preprocessing as train dataset. Hence, the train dataset may provide better predicted forecasting projection over the retracement levels. Thus, the evaluated ML and DL model provide better forecast with minimized the error rate of stock price prediction. This stock market analysis begins in the analytics area of Delta

Lake, which performs data analysis in a different order while Data Lake stores the information in its native state. The preparation process would not end until the data has been demanded by the applications or the query is run. Consequently, it contradicts the default ETL principle of Data warehouse. Instead of ETL, the Delta Lake encourages newer data processing order Extract-Load-Transform (ELT).

There is no predefined data schema in Delta Lake; information would be retrieved from a source and then transferred to the data lake depending on the requisite metadata. As a result, Delta Lake can manage both the write-intensive as well as read-intensive workloads, such as transactions and analytical, reuniting two contrasting requirements. The proposed architecture is used to address a variety of issues, including the collection of frequently requested data that arrives at a rapid pace and large data that arrives slowly. Azure databricks has been launched to deal with sophisticated analytic challenges as well as ML requirements. As a result, the data in ADLSG2 could be data sourced real-time or through batch processing.

II. Review of Literature

ADL is a versatile, adaptable and reliable framework. ADL is capable to store and analyze broad range of data as well as intended for huge workloads that require higher bandwidth. It is accessible using Storm, U-SQL, Hive, as well as Spark amongst other techniques. Microsoft's data lake solution consists of ADLS with Azure Data Lake Analytics (ADLA)
A simple architecture allows Data Lake to store information in native state. Every item in the lake is assigned an unique identification as well as an extensive metadata collection. It serves as a custom-built schema that the client creates depending on the query-related data, resulting in much smaller quantity of dataset which must be processed in order to provide answers to the customer's queries. There has been no set schema established previously as well as some queries on the chances of data becoming inconceivable have been discussed [11]. Hukkeri et al. provided a broad outline about different data lake offerings that are available in the industry for clients. Its architecture contains Hadoop, Amazon Web Services (AWS), as well as Azure data lakes. Apache Hadoop has been widely recognized as the industrial benchmark in Data lake. To address concerns about Hadoop's raw data as well as shortage of data security, both Azure Data lakes and AWS data lakes have built wrappers around it [12]. Snezhana Sulova had explained the concepts about data lakes as well as data warehouse and detailed their benefits and drawbacks. This review provides a high-level overview of the way Data Lake could be utilized resolve smart-grid data-management system challenges [13]. Surabhi D Hegde et al. had described the Lambda architecture of Data lake for data analytics. Evolving big data solutions include Data Lake and rapid data. Whereas both Data lake as well big data have the ability of storing as well as processing large information, rapid data provides insights in real-time based upon limited data [14].
C. Diaconu et al. details RSL-HK Ring infrastructure, which serves as the basis for tackling ADLS, that handles large file folders. The infra provides efficient, huge memory, expandable as well as persistent environment, all of that are key parameters of metadata operations. This infrastructure's deployment entails a new Paxos combination and the current transactional in-memory block data management design. The ability to dynamically add novel Paxos rings and accumulating machines to the existing Paxos ring determines the availability and scalability of the RSL-HK rings. Similarly, the RSL-HK Paxos elements rely upon this Cosmos deployment, which has delivered extremely high reliability over several years across hundreds of rings deployed for production. As a result, SQL Hekaton employs the leverage technology for transactional in-memory block management [15]. Delta Lake's legal implementation is the agile Business Intelligence (BI) fundamental principle devised with aim to provide precise data in the appropriate instance for precise analysis procedures. As a result, initial stage in a product development project in an organization has reduced errors and costs in subsequent stages [16]. The following are the key requirements for efficient deployment of Delta Lake data source [17, 18].

1) Clarify the target and requirements for the use of DeltaLake.
2) Develop and implement an information management strategy for Delta Lake.
3) Develop protocols to data access security as well as data control, along with effective data utilization
4) Create newer as well as broader data sources for analytics rather than relying solely on BI infrastructure elements.

The declarative edition provided with subsequent data frame APIs is used within the libraries which map the data preprocessing and compute into the Spark SQL query plan and benefit from customizations in both the Delta Lakes and Delta Engines. The technique works with Spark data frames and Koalas [19]. The most recent data frame API for Spark established by the Koalas improves the interoperability with Pandas. As a result, enterprises are now developing a wide variety in ML-specific data versions. The feature store system aids to redeployment of standard capabilities of DBMS. This makes it much easy to use Data Lake abstraction with the in-built DBMS functions which aid in the implementation of feature store functionality. Concurrently, declarative ML algorithms such as quantized ML may perform much better than any Lakehouse [20]. As opposed to scanning through native files of data lake, cloud-native DBMS offer a remedy as serverless processors which aid to combining of rich levels of metadata management via Delta Lake, resulting in improved performance [21]. Through including this information within the same record log, this application ensures that Delta Lake could accrue as most latest information and store that to a version field automatically. Comparable log entries are being utilized as a relevant Delta as within the addition as well as elimination of all events which are atomically entered in the log. Every program could generate their own appId at random and obtain an unique ID, whereas incase with Spark Structured Streaming, the function used is the Delta Lake connector. [22].

III. Research Methodology

The purpose of this research is to present and describe statistical analysis tools in line with Fibonacci sequence with a specific prominence on price patterns generated by Fibonacci numbers. In this research, phase 1 has presented Fibonacci sequence along with the ratio for forecasting the price of each individual share. Characterizing of the potential benefits on financial markets and the attempt of identifying

the price patterns in the actual realities of stock market is conducted and evaluated with ML and DL techniques.

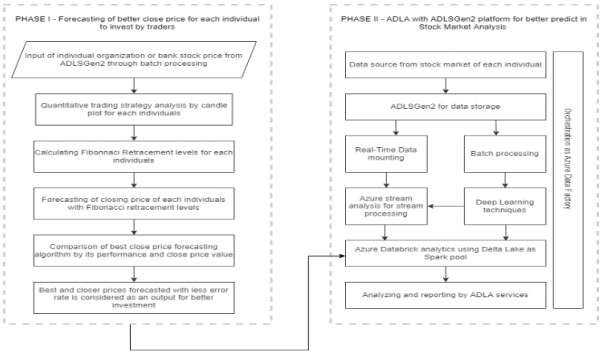

Fig. 1. Architecture of stock market analysis using ADLA with ADLSG2

Hence, forecasting of predicted price is evaluated through error rate performance and forecasted price of Fibonacci levels is determined along with less error rate. The analytical execution is progressed within the ADLA using spark pool. The Phase 2 illustrates the overall process of proposed big data platform with ADLSGen2 with ADLA for data storage and analytical executions over predicting of stock price by ML and DL technique which are progressed using bulk and Tera bytes data. The ADDL is used in this work to stream bulk and huge data without buffering or data loss. Figure 1 depicts the architecture introduced by the Azure Protocol for BDA, which is designed to keep data mounting, processing, and analysis of stock market price analysis.

A. Phase 1: Forecasting Fibonacci retracement levels by ML and DL techniques

Fibonacci retracements are popular among technical traders. It performs as a stock market tool to identify the trends and retracements in the stock prices. Each individual organization stock price is forecast to assist trader to decide their strategies of entry and exist for the specific stock in positional trading and for intraday. Most statistical investors utilize Fibonacci ratio to find out key level of support and resistance for deciding the opportunities of entry and exit. The ratios used for analysis are 0, 0.236, 0.382, 0.5, 0.618, 0.786, 1.0 and the settings of Fibonacci retracement initially requires to find out usual trends in the given time period. It is preferred to identify uptrends and downtrends. The low point in the candle plot dragged to high point is uptrend and dragging from high point to low point is downtrend. So, retracement levels can be listed generally based on low and high points. Based on investor types, various target profit levels are utilized, and this study have investigated the performances of each target profit level but for instance axis bank is shown in figure 2.

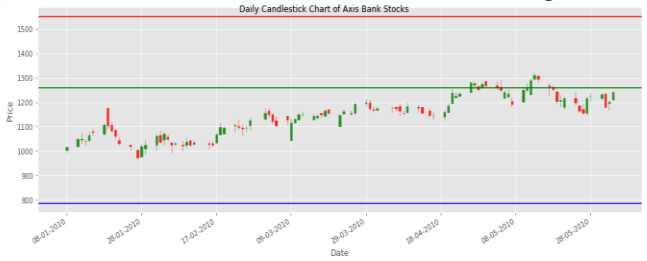

Fig. 2. Candle plot with fibonocci retracement for Axis bank Stock Price

The Golden Ratio is a special number whose value is 1.618 approximately which tends to reflect the principle of good design and structure. Fibonacci ratio is a mathematical formula that is expressed as a ratio and has the following values: 0%, 23.6%, 38.2%, 50%, 61.8%, and 100%. They used them to figure out key quantities that have an effect on an item. The Golden ratio of Fibonacci is 61.8%, however other ratios can be calculated, such as f78.6% = (1+5) / 20.5 0.786151. The ratios play a significant role in price reversal because the previous trend is likely to recede and return to one of the ratios as a result of deducting confirmed 0.214. To forecast trends in the stock market, futures, and commodities, the Fibonacci ratio is used in combination with other statistical analysis such as RSI, moving average, and others. The concept of trading off a trend is based on the retracement of the trend's Fibonacci support and resistance levels.

Steps involved in forecasting Fibonacci retracement levels

Taking into account the peak between extreme points as well as the horizontal lines drawn between the intervals of the second point, the Fibonacci ratios are 23.6%, 38.2%, and 61.8%. In contrast, the length interval-based Fibonacci numbers require the determination of two extreme points. As a result, it is critical to determine the zone of the Fibonacci expansion, such as the Fibonacci retracement, in order to achieve the best results.

Step 1: Initially collect the data source from ADLA and calculate Maximum (Max) price, Minimum (Min) price and the difference of Max and Min.

Step 2: Level of Fibonacci retracement is calculated in terms of Level1, Level2 and Level3. Level1 = Max-(Diff*0.236), for level2 with 0.382 and level3 with 0.618 are considered.

Step 3: The close price history is plotted with reference to 0% as Price_Min, 100% as Price_Max and 23.6%, 38.2% and 61.8% for the respective level1, level 2 and level 3.

Step 4: The original close price with levels is identified and set as levels of Fibonacci retracement level prices.

Step 5: Fibonacci retracement level price for overall dataset is obtained to understand the level of Fibonacci ratio. Dataset is preprocessed and split in term of Train dataset and Test dataset before forecasting by various ML and DL methods.

Step 6: The last four-year data is considered for test data and also considered as forecasted years for ML and DL models to understand the nearest close price of test dataset.

Step 7: The forecasting of ML and DL models with Fibonacci retracement level of close price is checked with original test data Fibonacci retracement level price. The performance each ML and DL model is evaluated with error rate performance.

B. Phase 2: ADDL has integrated azure protocol with databrick analytics using spark pool

Azure Databricks was introduced to address sophisticated analytical issues and the need for machine learning. ADLSG2 can store both real-time and batch processing of data. A batch layer is indeed a cold patch which keeps all of those received data as in its native format as well as executes batch processing. Every modification to the property of a particular data are registered as the latest event record along with a time-stamp, allowing the data collection to be re-computed at any point in time. Recompiling the batch view from the original data source is necessary because it enables the system to develop by including new features. Fully managed event

routing is provided by Azure Event Grid, which is configured as a service.

Furthermore, the serverless efficiency is delivered via Azure Function for the compute engine as well as Azure Logic Apps for the workflow management engine, that simplifies event-based processing and allows workflows as to respond to events in real-time. Because of the large number of data sources, this study focuses on batch processing. In general, big data solutions are achieved by processing data files with Long Running Batch Jobs (LRBJ).

The objective of the LRBJ is to capture, fitler out as well as arrange data for analytics. When an externally available data source has been developed, its use and relative path to the file are indicated in the Openrowset () function. As a result, data from different native source is stored in ADLSG2 via Azure Data Factory, which acts as an management tool as well as other tools for data movement. Events created by ADLSG2 for creating new file, updating files, modifying by deleting or renaming files are channeled using Delta Lake as the event grid and azure function by ADDL are connected. Figure 3 shows the new cluster created on the home page with event log by spark pool. This is done by hitting the "Create cluster" button, where the name of the cluster must be entered to particular cluster. Data Read could be sped up using Delta cache by creating remote file copies in the node's local storage utilizing fast intermediate data format.

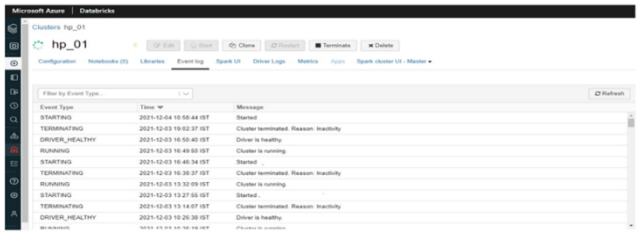

Fig. 3. New cluster generated in Azure databricks

Once files are pulled from the remote location, the data is automatically cached. Following comparable data reads executed at local lead to significantly improved reading speed. As a result, after clustering, the data from the respective CSV files is read into the data frames. Delta cache can read as well as execute data faster than Spark cache due to the use of efficient decompression protocols. ADDL and ADLSG2 collaborated on this study to provide a manageable service for large-scale cloud-based data warehousing. It enables users to analyze data using ADDL with ADLSG2 as a data modeling layer that could be able to execute multi-dimensional OLAP cube models over Azure Analysis services. Big data solutions include replicated data processing operations contained in workflows, with sinks loaded with processing data into an analytical data store. It modifies source data and moves it between sources in order to project results into a report. This is employed to automation of workflows through utilizing of orchestration technology such as Azure Data Factory. ADDL performance is evaluated using existing big data platform parameters such as memory usage, CPU usage, and load time.

IV. Results and Discussion

This prototype databricks runtime deployment is carried out with Apache Spark APIs that are open-source, utilizing a significantly improved processing engine which generates a great efficiency when comparing with the standard source of Apache Spark offered in the ADLS environment. If necessary, the default basic pushdown method is used in SparkSQL to define filters. As a result, the query processing stage is validated by assigning an explain command to the query.

TABLE I. Comparison of Original Close Price with Various Forecasted Techniques

Fibonacci Levels (%)	Original Close Price	Forecasted from ARIMA	Forecasted from KNN	Forecasted from STM	Forecasted from GRU
0	822.8	403.158	2019.825	863.446	988.667
23.6	717.297	142.562	1548.604	757.87	899.651
38.2	652.027	-18.655	1257.086	692.557	844.582
61.8	546.523	-279.25	785.864	586.982	755.566
100	375.75	-701.063	23.125	416.093	611.482

When a query plan includes a PushedFilter, the query is simplified to select the necessary data based on whether the predicate returns True or False. If there is no PushedFilter in the query plan, it is cast with the where condition. Fibonacci retracement level calculation in analytics is performed and forecasting of test close price with various ML and DL techniques are compared with original price of test data. This study illustrate with axis bank is shown in Table 1.

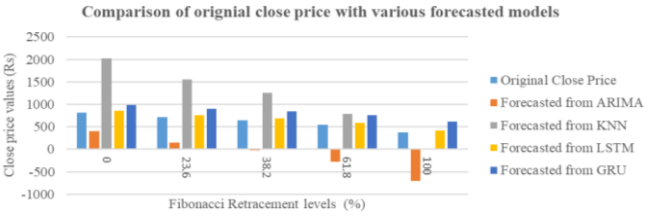

Fig. 4. Comparison of original close price with various forecast techniques

Figure. 4 illustrates comparison of forecasting the close price in actual realities of stock market. Fibonacci retracement level calculation is achieved and justified based on evaluation of forecasting DL and ML techniques. The LSTM technique has obtained nearest price with original close price value than the other forecasted models like ARIMA, KNN and GRU.

TABLE II. Comparison of Error Rate Parameter for Various Forecasted Techniques

Error Rate Parameters	Forecasted from ARIMA	Forecasted from KNN	Forecasted from LSTM	Forecasted from GRU
MAE	732.634	380.204	41.585	200.959
MAPE	732.634	380.204	41.585	200.959
RMSE	838.331	433.725	43.879	202.016

The negative price is obtained in KNN model that shows the downtrends in Fibonacci retracement level prices. Hence, the KNN is not suitable to this dataset with Fibonacci retracement levels. The model performance for forecasting is analyzed using error rate parameters like Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) as well as Root Mean Square Error (RMSE). Accuracy and noise can be determined through error rate parameters shown in Table 2.

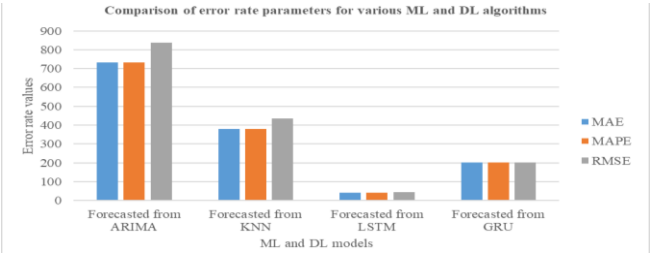

Fig. 5. Comparison of original close price with various forecast techniques

Figure.5 illustrates the comparison of various Fibonacci retracement levels based on predicted close price by forecast techniques evaluated through error rate parameters. The evaluation of forecasting techniques shows that LSTM technique has attains less error rate with 41.585 MAE and MAPE and 43.879 for RMSE in forecasted price than the other forecasted models like ARIMA, KNN and GRU. The analytics process is executed and transformed through spark pools. This experiment is carried out in a cloud infrastructure using the Azure protocol and a system configuration of 6 cores and 8GB of RAM. It has incorporated the ADDL with ADLSG2 for BDA, wherein its performance could be measured utilizing the Quality of Service (QoS). It was discovered that traditional ADLSG2 Cpu usage and memory usage is higher than ADDL with ADLSG2.

TABLE III. COMPARATIVE OUTCOME OF OPTIMIZATION IN ADDL WITH ADLSG2 VS TRADITIONAL ADLSG2

Date Time	CPU Usage in MHZ		Memory Usage in KB	
	ADDL with ADLSG2	*Traditional ADLSG2*	*ADDL with ADLSG2*	*Traditional ADLSG2*
2016-02-02 00:00:00	58.00	81.24	116763	135286.8906
2016-02-02 00:05:00	106.49	123.05	333189	406408.00
2016-02-02 00:10:00	61.23	68.06	148478	163492.1875
2016-02-02 00:15:00	48.47	65.72	128810	151165.3906
2016-02-02 00:20:00	59.71	64.16	116280	140813.1718

It assist to conduct improved stock market assessment and consume less memory. The envisaged ADDL with ADLSG2 architecture is capable of meeting a suitable requirement in different types of distinct data frames. As a result, ADDL with ADLSG2 method produced analytical and storage could be evaluated with metrics by load time and produce lesser CPU as well as memory utilization. The metrics show the utilization has been gradually scaled down the cost of data usage. Table 3 shows the QoS parameters for such recommended ADDL with the ADLSG2 also with optimization results for a particular date and time data analysis.

Memory usage

Figure 6 depicts memory utilization of ADDL along with ADLSG2 as well as conventional ADLS with a particular date and time chosen at random. The date and time interval is set as 5 minutes, and the consumer's memory utilization for the relevant data source is relatively low in ADDL with ADLSG2 compared to conventional ADLSG2

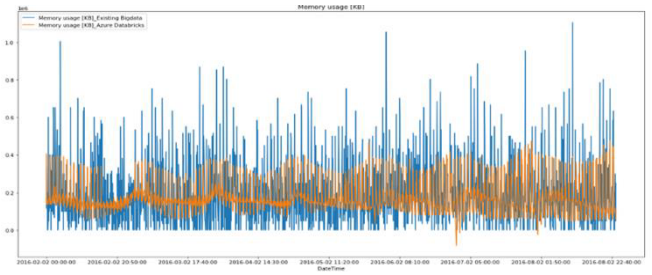

Fig. 6. Memory Utilization of AZURE Databricks Vs Traditional ADLS

The maximum memory usage of ADDL with ADLSG2 and traditional ADLSG2 at 2016:02:02 00:05:00 are 333.19 MB and 406.41 MB respectively. Memory usage by ADDL with ADLSG2 is reduced, allowing for more operations in read and write. Figure 7 depicts the CPU utilization of ADDL with ADLSG2 as well as conventional ADLS when a particular date and time chosen at random. The time and date interval is assumed per each 5 minutes, and the consumer's CPU utilization as for the relevant data source is comparably low in ADDL with ADLSG2 than in conventional ADLSG2.

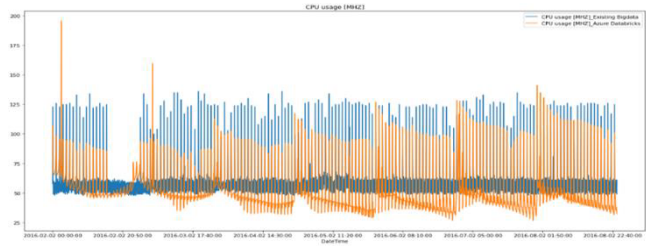

Fig. 7. CPU Utilization of AZURE Databricks Vs Conventional ADLS

At 2016:02:02 00:05:00, the maximum CPU usage of ADDL with ADLSG2 and traditional ADLSG2 is 106.49 MHZ and 123.05 MHZ, respectively, indicating that CPU utilization of ADDL with ADLSG2 is lesser and advantageous for analyzing and computing additional operations.

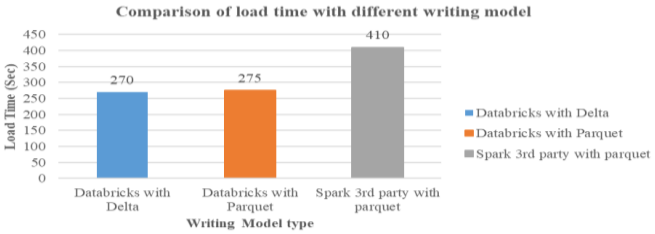

Fig. 8. Load Time comparison with various Writing model types

The Delta statistics accumulation has amassed the necessary overloads. TPC-DS has been employed for evaluating query response time in a single user mode. Likewise, this study concentrates upon load time with 250 GB and was analyzed using the TPC-DS store sales table formatted into CSV files on a cluster with one i3-2xlarge master system and six i3-2xlarge workers. The average of two runs and Spark pool performance in writing to Delta Lake is comparable to writing to Parquet, indicating that the statistical collection do not incur substantial overhead in other data loading work, as illustrated in figure 8. The operation of ELT is entirely comparable to the data set, and disk I/O is usually finetuned at the engine level to achieve high bandwidth.

V. CONCLUSION

This research work focus on stock market analysis initiated through quantitative trading strategy analysis by candle plot for progressing each individual using Fibonacci retracement levels. The forecasted LSTM technique attains best and closer price with less error rate and is considered as an output for better investment than other forecasted ARIMA, KNN and GRU model. Delta Lake is deployed as a storage medium as well as provides clients with the access protocol sets that are simple to use and highly available. This gives the client direct high-bandwidth connection to the object store. The use of Delta Lake including an ACID table as the storage layer upon a cloud object store offers a broad variety of DBMS efficiency along with data management while reducing cloud storage costs. The ADDL architecture along with the ADLSG2 has resulted thus in a storage having both reading as well as writing capabilities and an ELT processing upon the data source. The performance of ADDL with ADLSG2 is compared to traditional ADLSG2 using metrics such as CPU usage as well as memory consumption. ADDL along with ADLSG2 has low CPU and memory utilization, and TPC-DS is used to calculate load time and to determine the write performance of ADDL while in comparison to parquet and third-party parquet. The writing performance of ADDL demonstrates that the analysis loading time is improved for conducting BDA towards the stock market as with effective CPU and memory usage at a lower cost.

REFERENCES

[1] D. Chong and H. Shi, “Big data analytics: A literature review,” J. Manag. Anal., vol. 2, 175–201, 2015.

[2] V.S. Tomashevskaya, and D.A. Yakovlev, “Research of unstructured data interpretation problems,” Russian Technological Journal, vol. 9, no. 1, pp. 7-17, 2021.

[3] F. Cappa, R. Oriani, E. Peruffo, and I. McCarthy, “Big data for creating and capturing value in the digitalized environment: unpacking the effects of volume, variety, and veracity on firm performance,” Journal of Product Innovation Management, vol. 38, no. 1, pp. 49-67, 2021.

[4] C. Yang, q. Huang, z. Li, k. Liu, and F. Hu, “Big Data and Cloud computing: Innovation Opportunities and Challenges,” int. J. Digit. Earth, vol. 10, pp. 13–53, 2017.

[5] Pwint Phyu Khine, Zhao Shun Wang, Data lake: a new ideology in big data era, ITM Web of Conferences 17, 03025, 2018.

[6] N. Miloslavskaya and A. Tolstoy, Big Data, Fast Data and Data Lake Concepts, 7th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2016.

[7] D. Ilin, E. Nikulchev, “Performance Analysis of Software with a Variant NoSQL Data Schemes” 13th International Conference Management of large-scale system development"; IEEE, pp. 1-5, 2020.

[8] P.-N. Sawadogo, et. al, “Metadata Systems for Data Lakes: Models and Features,” 1st International Workshop on BI and Big Data Applications (BBIGAP@ADBIS 2019); Bled, Slovenia, pp. 440–451, 8 Sep 2019.

[9] P.P. Khine, and Z.S. Wang, “Data Lake: a new ideology in big data era,” ITM Web of Conferences, vol. 17, p. 03025, 2018.

[10] M. Armbrust, T. Das, et.al Delta Lake: High-performance ACID table storage over cloud object stores. In VLDB, 2020.

[11] Inmon, B.: Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump. Technics Publications (2016).

[12] Tanmay Sanjay Hukkeri, VanshikaKanoria, Jyoti Shetty, A Study of Enterprise Data Lake Solutions, International Research Journal of Engineering and Technology (IRJET) Volume:07 Issue:05 | May 2020.

[13] Snezhana Sulova, The Usage of Data Lake For Business Intelligence Data Analysis, Conference Paper · October 2019.

[14] Surabhi DHegde, Ravinarayana B, Survey Paper on Data Lake, International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064, 2016.

[15] C. Diaconu, et. al, Hekaton: SQL Server’s Memory-Optimized OLTP Engine. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pages 1243-1254.

[16] Nacheva, R. Principles of User Interface Design: Important Rules that Every Designer Should Follow. Izvestia, Journal of the Union of Scientists – Varna. Economic Sciences, pp. 140– 149. 2015

[17] Khine. P, and Wang. Z, Data lake: a new ideology in big data era, ITM Web of Conferences 17, WCSN 2017.

[18] Yordanova. S and Stefanova. K, Big Data Challenges – Definition, Characteristics and Technologies, The Scientific Papers of UNWE, 1, pp. 13-31, 2019

[19] M. Armbrust, et. al Spark SQL: Relational data processing in Spark. In SIGMOD, 2015.

[20] S. Li, L. Chen, and A. Kumar. Enabling and optimizing non-linear feature interactions in factorized linear algebra. SIGMOD, page 1571–1588, 2019.

[21] M. Perron, R.Castro Fernandez,D. DeWitt, and S. Madden. Starling:A scalable query engine on cloud functions. SIGMOD, pg131–141, 2020.

[22] M.Armbrust, et. al. Structured streaming: A declarative API for real-time applications in Apache Spark. SIGMOD, pg 601 – 613, 2018.