International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-12 Issue-3, February 2023

Template-driven Real-Time Data Acquisition Technique with Big Data

Sowmya R, Suneetha K R

Abstract: ***The development of Big Data (BD), which is used to obtain numerous data from various domains, is brought about by technological advancement. However, managing the information and extracting knowledge from it is the most challenging and problematic. Thus, this paper proposed a template-centric new Data Acquisition (DAQ) methodology. The stock market data is gathered from several structured or unstructured data sources. After the DAQ criterion, templates are created for the gathered data. The stock market data is collected based on its Application Programming Interface (API) and transmitted via the transmission protocols during the DAQ process. To effectively remove redundant data, the transmitted data is pre-processed and stored efficiently in the network for further real-time analysis. Finally, the performance of the proposed technique is evaluated. According to the experimental and empirical evaluations, the proposed system outperforms the other methods.***

Keywords: Big Data, Pre-processing, Decision Trees, Data reduction, Cleansing, Data Integration, Real-Time.

I. INTRODUCTION

BD technology, which harnesses the potential value of data to generate massive economic benefits, can be leveraged for data mining and analysis, as well as the sharing of huge datasets, as it is one of the critical development directions in the information field. In the meantime, for social and economic development, it could offer decision-making strategies (Wang, 2020, [1]). BD, which is an evolving phase, encompasses a fusion of (A) structured, (B) semi-structured, along with (C) unstructured data that pose a tedious task to be processed by employing traditional methods and databases (Juneja, 2019, [2]). (1) Business transaction systems, (2) customer databases, (3) mobile applications, (4) websites, and (5) machine-generated data, together with (6) real-time data sensors, are the various sources from which the BD could be evolved. Volume, variety, velocity, veracity, and value are the five differentiating qualities explored by voluminous data (Majeed, 2021, [3]). The volume aims at the data set size, which can range from megabytes to gigabytes, terabytes, or even petabytes, depending on the quality of the BD. Various configurations, layouts, and kinds of data essential to support are termed variety (Mohamed, 2020, [4]).

Manuscript received on 03 January 2023 | Revised Manuscript received on 09 January 2023 | Manuscript Accepted on 15 February 2023 | Manuscript published on 28 February 2023.
*Correspondence Author(s)
Sowmya R*, Research Scholar, Department of Computer and Engineering, Bangalore Institute of Technology, Bengaluru, Visvesvaraya Technological University, Belagavi (Karnataka), India. E-mail: sowpu29@gmail.com ORCID ID: https://orcid.org/0000-0002-4527-8060
Dr. Suneetha K R, Professor, Department of Computer and Engineering, Bangalore Institute of Technology, Bengaluru, Visvesvaraya Technological University, Belagavi (Karnataka), India. E-mail: suneetha.bit@gmail.com

The velocity signifies the speed of data generation along with dissemination. Another feature of BD, where trustworthiness is the primary goal, is termed velocity. The usefulness of data, associated with its characteristics of veracity, is determined by its value. Thus, the more prominent the data quality, the more valuable it is (Neilson, 2019, [5]).
Some valuable and helpful information could be extracted if such BD is extensively evaluated (Dai, 2020, [6]). Data gathering is the process of collecting information from multiple data sources. As incorrect data collection will impact future data analysis, it must be compelling and legitimate (Rohini, 2022, [7]). Recently, real-time-centric applications have been widely deployed, whose processes require real-time processing of information for data visualisation and learning techniques to be utilised. Since the input data are continually engendered at each instant, systems functioning on the data's real-time processing have to be quicker; in addition, outcomes have to be attained in parallel (Munawar, 2020, [8]). Thus, processing large volumes and varieties of data for extracting meaningful information becomes crucial in making data-driven decisions (Bhattarai, 2019, [9]). Instant and efficient data processing is a significant ability in the BD domain. At the same time, it's usually tedious to process such extensive data to attain the essential services provided by the domain in traditional techniques (Arooj, 2022, [10]).

A. Problem Definition

Due to the following limitations, the BD lacks high data throughput and strict real-time requirements, despite encompassing numerous techniques and methods.

- Various challenges are caused by the complex combination of data and time in BD in real-time to meet the demands on a large scale.
- Problems like high computational cost and algorithmic instability are generated by high dimensionality fused with a large sample size.
- Moreover, data representation, as well as unified models, are not explored by the prevailing methodologies.

According to the above requirements, generating a template-based configuration file involves a data acquisition and analysis system. The proposed system's key contributions are,

- To capture the data from different sources in real-time, a template-driven approach is introduced.
- To present generic data capture via dynamically configured APIs
- Collecting the same data from multiple sources and removing the redundancy to achieve good results with prediction.

The Block diagram helps explain the system's representation more elegantly.

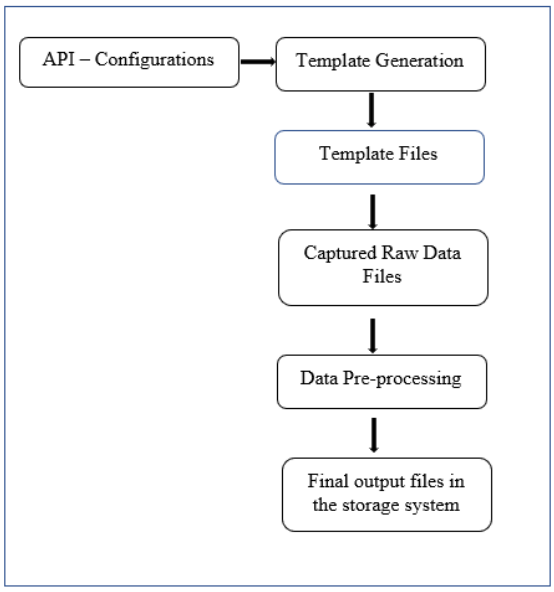


Figure 1: Block Diagram of the proposed methodology.

The remaining part is arranged as follows: previous data-gathering techniques are detailed in Section 2, the proposed DAQ system is elucidated in Section 3, the experimental results obtained by the proposed framework are presented in Section 4, and the paper concludes with a discussion of future work in Section 5.

II. RELATED WORK

(Jindal, 2020, [11]) developed a tensor-centric BD management methodology for the data collected from the Internet-of-Energy (IoE) environment. The tensor operations, using higher-order singular value decomposition, extracted the critical data from the IoE. To offer several services in smart cities, dimensionality-reduced data was employed. According to the results, this system demonstrated superior performance compared to traditional techniques. However, this system incurred computation overhead to form tensors from the collected data.

(Osman, 2019, [12]) explored a BD analytics system termed "Smart City Data Analytics Panel – SCDAP." New functionalities were presented in the BD analytics, depicted in data model management and aggregation. The values were weighed against the traditional knowledge discovery systems. The ability to deal with multiple platforms was constrained, as the separation between SCDAP functionalities and the underlying data storage and management layer had to be maintained.

(Mavrogiorgou, 2019, [13]) exhibited an innovative system to assess the data quality in correlation with the data source quality. A 5-step system was deployed, which recognised, identified, and linked the present data sources with the collected data. The acquired data was cleaned and correlated with the quality measurements to detect the qualified data. According to the experimentations, effective outcomes were achieved in accessing data sources and ensuring quality. However, this system was constrained to information.

(Geng, 2019, [14]) explored an industrial BD platform for minimizing the data processing time while requiring less data storage space. The goal is to analyze the effect of various compression and serialization techniques on BD platform performance. It also aims to select the optimal compression and serialisation methodologies for the industrial data platform. According to the results, when compared to the prevailing techniques, it took less time with the increased data. The system was slow when the raw data was added to the index. (Jimenez-Marquez, 2019, [15]) developed a 2-stage system. The primary goal was to prepare data and detect an optimal Machine Learning (ML) system. The next stage depended on the introduced layers of BD, which aimed to attain the desired result by including most of the ML model from the initial step. Various ML techniques were trained for 2, 3, and 5 classes, with the best outcomes detected for binary classification. By registering an enterprise API, historical tweets were only available for a maximum range of '3' weeks. (Hernandez-Suarez, 2018, [16]) developed a new system to gather historical tweets within any date range by employing web scraping bypassing Twitter API restrictions. for acquiring an unlimited volume of tweets bypassing date-range limitations, the power of scraping engines was augmented by using Scrapy, which is an open-source together with the collaborative system to extract data as of websites written in Python. Over long periods, Python technologies generate massive amounts of data with faster outcomes. However, it could not ingest data in real-time.

(Kaur, 2018, [17]) explored a tensor-based BD management framework for the dimensionality reduction problem of BD generated by enormous intelligent devices. Initially, to minimise the reconstruction error of the minimised tensors, the Frobenius norm was implemented. An empirical probability-centric control scheme was developed to estimate an optimal path for forwarding data in a decreasing manner. According to the outcomes regarding experimental parameters, the system's efficacy was demonstrated. However, it lacks support for more widely used data formats.

(Chen, 2017, [18]) presented a BD platform for satisfying the Mayo Clinic's (MC) daily healthcare requirements for HL7 message processing. Two identical Hadoop clusters– each with an Elastic Search cluster and instances of a storm topology – Mayo Topology for processing HL7 messages on MC ESB queues into an Elastic Search index, and the HDFS are included. According to the results, the applied BD system met or exceeded MC enterprise-level patient care requirements. Nevertheless, the nodes cannot be accessed without proper managerial approval, which requires a human or an application program. (Faheem, 2021, [19]) provided a cross-layer data-collection system termed CBI4.0 meant for active monitoring and control of manufacturing processes in Industry 4.0. In the Multichannel Wireless Sensor Networks (MWSNs), the CBI4.0 scheme exploited the sensor network's multi-channel and multi-radio architecture by dynamically switching between diverse frequency bands. According to the results, this system outperformed the existing methodologies. However, the system didn't include data from other data sources; instead, it focused solely on sensor-captured data. (Liu, 2021, [20]) presented Distributed Storage and Calculation Scheme for Long Time Series (DSCS-LTS) and a Correlation Coefficient Calculation Algorithm for Long Time Series (CCCA-LTS) in a distributed environment.

For visualisation, the granularity selection, along with the query process logic, was deployed. According to the experimentations, the system was effective, and data was managed by utilising actual data. However, correlation doesn't consider the effect of other variables outside the two being explored.

(Henry, 2021, [21]) developed Twi Scraper, a collaborative project and a module permitting user-centric data collection: Twi-FFN. To improve Twitter data collection, the collaborative project TwiScraper employed scraping techniques. To collect a user's network more efficiently when compared to the official Twitter API, the Twi-FFN module was developed. The community tool enabled the network to collect Twitter users in parallel. The efficacy was lowered since this system failed to capture the usual updates of every TwiScraper module.

(Mendhe, 2021, [22]) explored a BD analytics platform, which allows the user to focus their Twitter search criteria and attain access to massive Twitter Data at the touch of a button. The system supported social media data collection, and several filters were implemented for data cleaning, with further usage in ML and Artificial Intelligence-centric systems. For healthcare-related issues and research, this system depicted high levels of effectiveness. As the queries exceeded rating intervals and time ranges, disadvantages took place.

(Sharma, 2020, [23]) deployed the Open Biomedical Annotator (OBA), an ontology-centric Web service. By employing MedlinePlus Health Topics and National Drug Data File (NDDF) ontologies, domain-specific knowledge was collected. The medical-related terms were filtered out and stored in a medical term database, then forwarded to the UMLS REST API, providing the most precise and comprehensive health terminology platform globally. Automating the pre-processing steps is strongly required to facilitate data collection, processing, and database creation.

(Shah, 2022, [24]) exhibited a solution to tackle the issues of real-time analysis by employing a configurable Elasticsearch search engine. For large-scale text mining, (A) a distributed database architecture, (B) pre-built indexing, together with (C) standardising the Elastic-search framework, were deployed. Large-scale data's real-time analysis was made efficiently by an apt configuration of Elasticsearch and Kibana; it also assisted policymakers in viewing the outcomes instantaneously. Due to low scalability, high installation costs were incurred.

(Tavares, 2019, [25]) applied Funneling Wider Bandwidth (FWB) augments schedule length in networks with radios operating with the extension. In estimating the minimal number of time slots, this system was optimal for achieving a higher average throughput with fewer time slots. According to the results, the number of packets received per second increased with the wider bandwidth emulated in the topology, compared to a single bandwidth. Owing to the generation of data at regular intervals from several nodes, data processing was a key challenge at the network level.

III. PROPOSED TEMPLATE-DRIVEN DATA ACQUISITION METHODOLOGY

Specific challenges arise due to the massive size and significant dimension of the vast data produced at a time in the network, despite BD providing various new services and opportunities to modern society. Thus, for an efficient stock market DAQ process, a new template-based DAQ technique is proposed. In Figure 2, the proposed system's architecture is depicted.

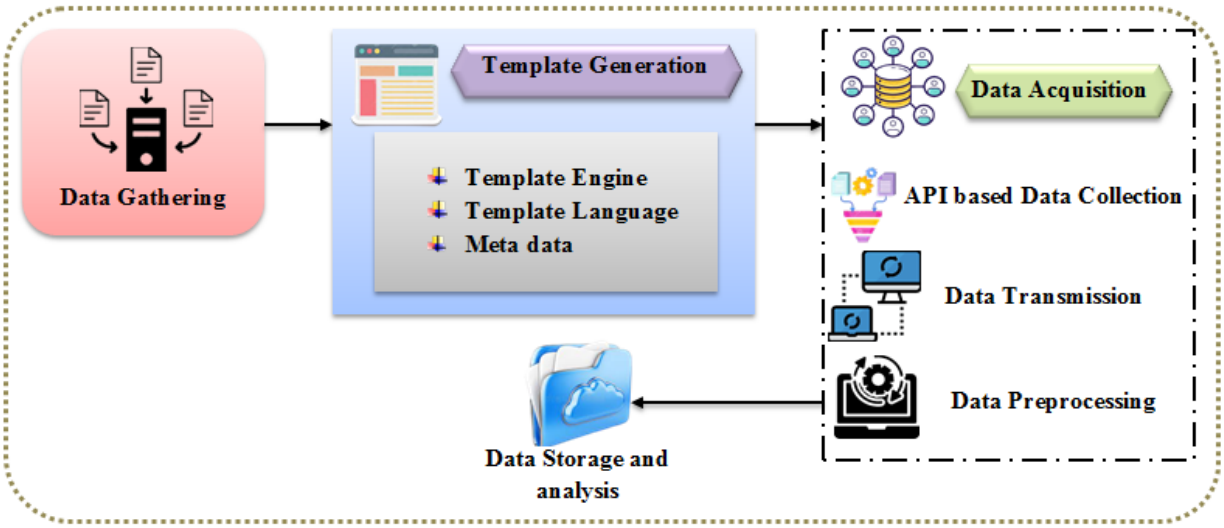


Figure 2: Architecture of the proposed methodology.

A. Data Gathering

Primarily, data gathering takes place to initiate the DAQ process. Data gathering involves rescuing raw stock market data from various structured and unstructured data sources. Typically, data acquisition (DAQ) is performed using multiple methodologies, which are categorised into pull-based and Push-based methods. Using dedicated data collection technology, data is obtained from various sources, including web APIs, Twitter APIs, and others. Thus, the data-gathering $\left(D^{g}\right)$ The phenomenon involved in the DAQ methodology is designed as,

$$D^{g} = D^{1}, D^{2}, D^{3}, \ldots, D^{G} \quad (1)$$

Here, the $G-$ The Number of gathered raw stock market data is signified as $g = 1,2,3,\ldots,G$.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

B. Template Generation

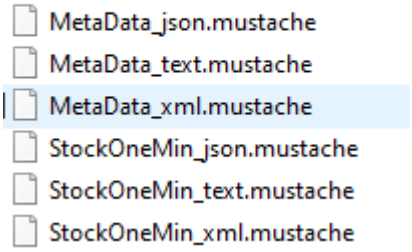
The gathered data enters the template generation stage. It is necessary to create a standard format for each type of data, referred to as a template, to facilitate a more efficient acquisition process, as the data collected from multiple sources is in various formats. Templates accept the data in any format and convert it into a standard form. For formatting and controlling the flexibility of the DAQ process, it is provided with if-then-else and for-loops control statements. It is necessary to create a template for the corresponding data to acquire new data from the data source. The template creation procedure is given as follows,

```
{
"{{model.className}}": [

{{listDataOpen}}
 { {{#items}}
  {{open}}#is_{{name}}{{display}}{{close}} "{{name}}" : {{templateValue}},
  {{open}}/is_{{name}}{{display}}{{close}}
  {{/items}}},
        {{listDataClose}}
]

}
```

Additionally, the generation of templates involves two primary components: a template engine and a template language. They are detailed further.

Template engine: It is primarily used to deliver server-side data to the corresponding application more efficiently. Using a template engine, the variables in the template file can be easily replaced with actual values during processing. EJS, Jade, Pug, Moustache, and Blade are the most commonly used template engines. Moreover, templates store the data in JSON or XML format. The deployed template file is delineated further,

Template files:



Template language: It refers to the programming language-agnostic semantics. The general syntax of the template language (T) is equated as,

$$T = \tau + \kappa \tag{2}$$

Where the timestamp is signified, and the configuration parameters like station, variables, and data collection technique are depicted κ. Hence, the probability of utilizing a template (T) For the given data (D^g) is equated as,

$$P(T \mid D^g) = \frac{\sum_{j=1}^{J}\sum_{g=1}^{G} P(\hat{T}^J \mid D^g)}{J} \tag{3}$$

Where the j^{th} sub-template of the J sub-templates in T is signified as T^j. Thus, the summation usage prevented the score value from reaching zero in the case of zero probability.

Moreover, the likelihood of using the sub-template $(\hat{T})$ For the given data (D^g) is equated as,

$$P(\hat{T} \mid D^g) = \frac{count(D^g, \hat{T})}{count(D^g)} \tag{4}$$

Here, the number of times the data (D^g) occurs at a particular time is exemplified $count(D^g, \hat{T})$, and the total occurrences of the data (D^g) are depicted as $count(D^g)$. The fluency score $(F_s(D^g))$ It is the rate at which the data enters the template.

$$F_s(D^g) = F(D^1, D^2, D^3, ..., D^G) \tag{5}$$

The above equation can be rewritten as,

$$F_s(D^g) = \frac{1}{G}\frac{\sum_{j=1}^{J}\sum_{g=1}^{G} P(\hat{T}^J \mid D^g)}{J} \tag{6}$$

Hence, the templates generated for the $G-$ number of data $(\tilde{T}_{(i)})$ are equated as,

$$\tilde{T}_{(i)} = \tilde{T}_{(1)}, \tilde{T}_{(2)}, \tilde{T}_{(3)}, ..., \tilde{T}_{(I)} \tag{7}$$

Where the $I-$ The Number of templates generated is signified as $i = 1,2,3,....,I$.

Metadata: After the template generation process, a metadata file is created. Simply structured data that aids in identifying the attributes related to the stock market data is termed metadata. Information not available in the template file is encompassed in the metadata. It is mainly comprised of semantic information. The resources related to stock market analysis are efficiently accessed and identified by disseminating

them through the use of metadata. The pseudo-code for the proposed template generation approach is cited below,

Pseudocode for the proposed template generation process

Input: Data gathered (D^g)

Output: Templates generated $(\tilde{T}_{(i)})$

Begin

Initialize the data gathered

For $1 \le g \le G$

Generate $T = \tau + \kappa$

For each g

Evaluate $P(T \mid D^g) = \dfrac{\sum_{j=1}^{J}\sum_{g=1}^{G} P(\hat{T}^J \mid D^g)}{J}$

Determine $P(\hat{T} \mid D^g) = \dfrac{count(D^g, \hat{T})}{count(D^g)}$

Calculate Fluency score $(F_s(D^g))$

End for

End for

Return Templates $(\tilde{T}_{(i)})$

End

C. Data Acquisition

DAQ occurs after the template generation criterion is met. The data is extracted by employing the standard template created for each source API in the previous step. The DAQ procedures are API-based data collection, transmission, and pre-processing. The process involved in the DAQ phenomenon is detailed as follows,

1) API-based data collection

The API is used to access data stored in a template format. An API is a computer programming interface that allows software to communicate with one another. This facilitates improvements in the usage of social media services, such as Twitter or Facebook, that are integrated with other social media services. Here, two types of APIs, namely external APIs and web-based APIs, are utilised. Regarding the data exchange process, the external API provides technical instructions in the form of a request for filtering, clarifying, and data delivery protocols. Then, the web-based APIs provide the data in a machine-understandable format. APIs are responsible for distributing the requests of web applications together with responses from servers by employing the Hypertext Transfer Protocol [HTTP]. RESTful APIs are referred to as Web APIs, which adhere to the Representational State Transfer (REST) architectural principles. REST is an approach that enables communication between numerous modules, most often used in the development of web services. The data extracted using the API is displayed below,

```
class PriceData:
    def __init__(self, open, high, low, close, volume):
        self.open = open
        self.high = high
        self.low = low
        self.close = close
        self.volume = volume
```

The mathematical formulation for the data collected using the API (A^p) is equated as,

$$A^p = A^1, A^2, ..., A^P \quad (8)$$

Here, the P – The Number of data collected using the API is signified as $p = 1,2,3,....,P$.

2) Data Transmission

For further processing of various real-time and analytical tasks, the data must be transferred to a reliable storage system upon successive collection. Here, the data centre transmission procedure is used. The data collected using the API (A^p) is transferred and stored in the data centre (D^c) In the data centre-based transmission technique. The data is retrieved from the data centre, processed, and utilised according to the transportation protocol as needed. The deployed transportation protocols are elucidated further.

Transportation protocol: TCP and UDP are the most used protocols for data transmission. Transmission Control Protocol is the expansion of TCP; in addition, it is mainly used for establishing and maintaining the data exchange. The User Datagram Protocol (UDP) is responsible for transmitting a large amount of data. Either TCP or UDP is used for data transmission based on the requirements. In the end, the data stored in the data centre uses the transportation protocol $(\hat{A}^p)$ is modelled as follows,

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$$\hat{A}^{p} = \sum_{p=1}^{P} A^{p} + n_0 \quad (9)$$

Here, the noise added during data transmission is signified as n_0.

3) Data Pre-Processing

Then, pre-processing takes place. The data is collected from multiple sources, and the stored data may contain unwanted, redundant, or noisy data, which can affect DAQ accuracy and storage space. One of the most crucial pre-processing steps involved in the proposed DAQ strategy is data de-duplication, which is discussed below.

Data Deduplication: It is adopted to eliminate duplicate copies of the data. According to the deduplication technique, each piece of data is assigned an identification in storage. It is considered redundant and is replaced with stored data when the data with the exact identification reaches the network. In the network, redundant data storage is highly discouraged due to the significant reduction in storage space. The pre-processing data phase is equated to,

$$\hat{D}^{g} = C^{0} + \hat{A}^{p} \cdot \chi^{ref} + \gamma \quad (10)$$

Here, the pre-processed data is signified as $\hat{D}^{g}$ The pre-processing coefficients are depicted as C^{0} The unmodeled part is mentioned, and the adjustment factor is delineated as χ^{ref}. The pre-processed data $\left(\rho^{m}\right)$ The Obtained is expressed as,

$$\rho^{m} = \rho^{1}, \rho^{2}, \rho^{3}, ..., \rho^{M} \quad (11)$$

Where, the M – The Number of pre-processed data is denoted as $m = 1,2,3,...,M$.

4) Data Storage and Analysis

For organizing the pre-processed data for efficient analysis and extraction purposes, the pre-processed data is then stored in the file system (template). Some of the data analysis techniques include reasoning, data mining, Semantic analysis, Machine learning, and information extraction, among others. Here, the Information Extraction (IE) technique is employed by utilising the corresponding API to extract the data stored in the template. For extracting structured stock market data from unstructured or semi-structured data, IE methods are employed. It is primarily used for the efficient processing of critical data, such as stock market data and medical reports, through entity recognition, linked data, and machine learning, among other applications. Some of the IE processes are elucidated further.

Data Exploration and Visualisation: For the effective acquisition of stock market data, the incoming pre-processed data is analysed and visually inspected using graphical means.

Data mining is the process of discovering patterns in stock market data.

Model representation: The data is represented by employing one or more independent variables via the regression method $\left(\Re\right)$ and is formulated below,

$$\Re = \frac{1}{1 + \frac{1}{e^{-\alpha}}} \quad (12)$$

Where the logistic function is signified as α and is equated as,

$$\alpha = \log\left(\frac{\Re}{1-\Re}\right) \quad (13)$$

Thus, the data extracted from the template using the API $\left(\varepsilon^{f}\right)$ is equated as,

$$\varepsilon^{f} = \sum_{m} \rho^{m} \cdot \varsigma, 1 \le m \le M, 1 \le f \le F \quad (14)$$

Where the adjustment factor is stimulated by ς, and the data extracted are depicted as $f = 1,2,3,..,F$.

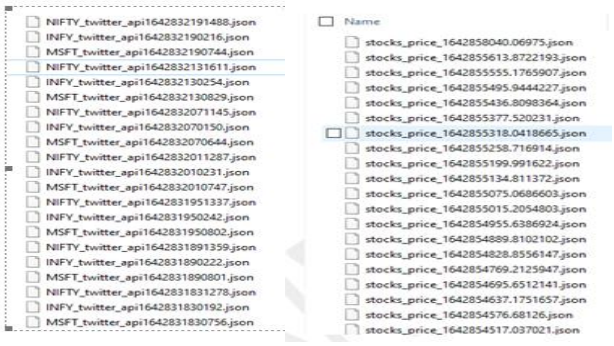
The pseudocode for the proposed template-based DAQ methodology is illustrated further

Pseudocode for the proposed methodology

Input: Stock market data $\left(S^{d}\right)$
Output: Extracted data $\left(\varepsilon^{f}\right)$
Begin
  Initialise structured and unstructured stock market data
  For each $\left(S^{d}\right)$
    Perform data gathering $\left(D^{g}\right)$
    For $1 \le g \le G$
      Generate templates $\left(\tilde{T}_{(i)}\right)$
    End for
    Acquire data $\left(A^{p}\right)$
    Transmit the acquired data
    Store $\hat{A}^{p} = \sum_{p=1}^{P} A^{p} + n_0$
    Perform pre-processing
    Obtain $\left(\rho^{m}\right)$
    Perform data analysis
  End for
  Extract data $\left(\varepsilon^{f}\right)$
End

IV. RESULTS AND DISCUSSION

Here, the functionality of the presented framework is tested by employing a setup with two input data sources: the External API and the Twitter web API. According to the defined schedule, the data files have been extracted every minute for the selected APIs in real-time. The Stock Market data are sourced from various data sources. In Figure 3, the sample output files created in the output folder using Twitter APIs and the external APIs are shown.



(a) (b)

Figure 3: Output files created at the output folder using (a) Twitter API and (b) External API

A. Performance Analysis

In this subsection, the performance of the proposed framework is evaluated using the Twitter API and an External API to extract data from various sources.

The amount of time taken to remove the output, in relation to the volume of data extracted, is used for comparison. The analysis is carried out for the number of output files extracted on parallel distributed nodes, such as P1, P2, and P3, as well as the time taken to extract the files. The P1 and P3 nodes have 8 GB of memory, with a CPU base speed of 2.70 GHz. The P2 node features 16 GB of memory and a CPU base speed of 2.95 GHz. The number of extracted files by the system is plotted as follows,

Table 1: Performance analysis using Twitter API

Nodes	Data Entities	Output files	Time Taken (Sec)
P1	50	50	130
P2	50	50	140
P3	50	50	130

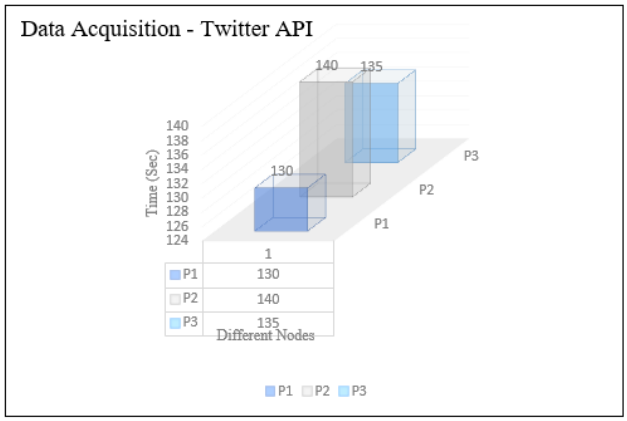


Figure 4: Performance analysis using Twitter API for nodes P1, P2, and P3

Table 1 analyses the performance of the proposed template-driven approach using the Twitter API. From Table 1, it can be seen that the nodes have retrieved 50 output files for the 50 data entities. Figure 4 shows the analysis in graphical representation. However, the time taken by the nodes varies depending on the number of files and the different periods under other loads. The P1 and P2 nodes retrieved the files within 130 seconds, whereas node P3 took an additional ten seconds to recover the output files.

Similarly, the output files extracted using the external API are discussed further as,

Table 2: Performance Analysis using External API

Run ID	Data Entities	Output files	Time Taken(sec)
P1_1	50	1	120
P1_2	50	1	140
P1_3	50	1	120
P1_4	50	1	130
P1_5	50	1	60
P1_6	50	1	120

When using an external API, the output files extracted from node P1 with different run IDs are analysed in Table 2—the Run IDs range from P1_1 to P1_6. For the data entities 50, one output file has been extracted using all run IDs. Concerning time, the run ID P1_5 took significantly less time. In comparison, P1_2 consumed substantially more time for removing the output file.

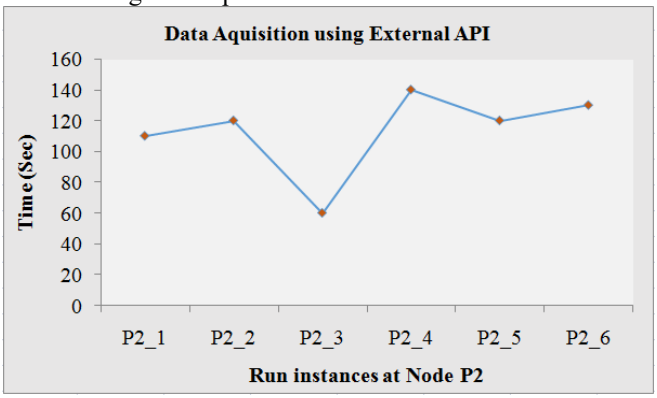


Figure 5: Performance analysis using the external API for node P2

The time taken by the Run IDs P2_1 to P2_6 to extract the output files from node P2 is shown in Figure 5 above. In analyzing figure 5, the Run ID P2_3 took only 60 seconds to remove one output file from 50 data entities. However, P2_1 took 50 seconds, while P2_2 and P2_5 took 60 seconds, and P2_4 and P2_6 took 70 to 80 seconds longer than P2_3.

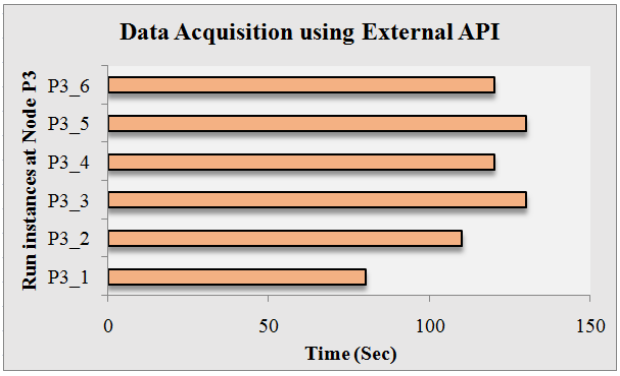


Figure 6: Performance analysis using the external API for node P3

Figure 6 analyses the time required to extract the output file for the given data entities associated with node P3. The research shows that Run ID P3_1 performs better, with a time of 80 seconds, which is 30

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

seconds less than P3_2, 40 seconds less than P3_4 and P3_6, and 50 seconds faster than P3_3 and P3_5. The study shows that the Run ID P3_1 performs better for node P3. The Run ID P3_1 results in less time, whereas the Run IDs P3_2 to P3_6 consume more time.

Based on the above results and analysis, performance parameters can be derived as follows:

1. Scalability: The template creation provides the flexibility to configure APIs dynamically. The output files are independently generated at the individual entity level. Scaling out the workload by increasing the number of execution instances is a significant advantage, as it helps utilise resources at the machine level more efficiently.
2. Response time: If there are 5000 different indices in a market. Suppose there are 500 other instances configured, each with 50 separate entities. Then, output files are created for all 5000 indices with the same 't+i' time seconds, where i is the small interval.
3. Processing time: If 'r' resources in a system extract x The number of output files at time t is then (p * x) output files are extracted on p parallel, distributed nodes, such that the time taken remains t or is slightly more than t.
4. Data Variety: Data variety is one of the significant challenges in Big Data. With the current framework, the various APIs and data sources enable the retrieval of information for the same entity from different sources in varying formats.

Next, the proposed framework is benchmarked against the leading existing data-capturing tool on the market through an analysis. It combines messaging, storage, and stream processing to allow storage and analysis of both historical and real-time data. One of the top tools for Data capture - Apache Kafka, is primarily used to build real-time streaming data pipelines and applications that adapt to the data streams.

Features	**Benchmarking – Kafka** (Le Noach, 2017, [26]) (Y Roh, 2021, [27]) (S Vyas, 2021, [28])	**Proposed Framework** [Sec. IV]
Scalability	Yes	Yes
Flexibility & Response time	Yes	Yes
Data Variety	Yes, it supports logs and IoT sensor data.	Yes, limited to data-driven from APIs.
Processing time	Yes	Yes
Extra tech experts for tuning up and monitoring the system	Required	Not Required
Dependency on the third-party tool – to store the metadata	Yes, Zookeeper	Not required.
Complexity – with setting up and deployment	Yes	No
Technical difficulties	Yes, message tweaking and wildcard topic selection	No
Monitoring tools	Not available	Yes – debug logs
Suitable for startups and small-scale industries	Concerns are there	Best

V. CONCLUSION

This paper developed a framework for data acquisition in big data. The framework primarily consisted of template-driven approaches for collecting data from various sources, including Twitter and Web APIs. The Data source can be configured dynamically with the supporting input and output files. To evaluate the performance of the proposed framework, experiments were conducted using stock market data obtained through the Twitter API and an External API. The results and the benchmark analysis revealed that the proposed framework is more efficient and the best attempt to deal with Big Data for real-time data analytics, with the following:

- Compliance with all the basic requirements of the Data collection process.
- Framework for the collection that includes security, compliance, and governance from the start.
- Tune and tweak data collection and data governance as use cases emerge and the data program matures, identifying what data sets are missing from the organization's extensive data collection process and what collected data sets hold no value.
- Automate the process as much as possible from data ingestion to cataloguing to ensure efficiency and speed
- Suitable for transactional data and structured/unstructured data.
- Easy to use and deploy for start-ups and small-scale industries.
- Independent module to collect raw data without any other data-related processes, focusing on essential information.
- Supports storing raw output files in VM, external storage, Hadoop Distributed File System, and cloud storage.

In the future, the data collection and analysis framework can be utilised in machine learning-based studies and extended to provide new opportunities for enhancing the template approach across different data sources in Cloud computing.

DECLARATION

Funding/ Grants/ Financial Support	No funding.
Conflicts of Interest/ Competing Interests	No Conflict of Interest.
Ethical Approval and Consent to Participate	Not Applicable.
Availability of Data and Material/ Data Access Statement	Data extracted from Twitter AP and AlphaVantage.
Authors Contributions	The first author serves as the corresponding author, while the second author is a member of the review team.

REFERENCES

1. Wang, J., Yang, Y., Wang, T., Simon Sherratt, R., & Zhang, J. (2020). Big data service architecture: A survey. *Journal*

of Internet Technology, *21*(2), 393–405. https://doi.org/10.3966/160792642020032102008
2. Juneja, A., & Das, N. N. (2019). Big Data Quality Framework: Pre-Processing Data in Weather Monitoring Application. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives and Prospects, COMITCon 2019*, 559–563. https://doi.org/10.1109/COMITCon.2019.8862267 [CrossRef]
3. Majeed, A., Zhang, Y., Ren, S., Lv, J., Peng, T., Waqar, S., & Yin, E. (2021). A big data-driven framework for sustainable and innovative additive manufacturing. *Robotics and Computer-Integrated Manufacturing*, *67*,1-21. https://doi.org/10.1016/j.rcim.2020.102026 [CrossRef]
4. Mohamed, A., Najafabadi, M. K., Wah, Y. B., Zaman, E. A. K., & Maskat, R. (2020). The state of the art and taxonomy of big data analytics: View from a new big data framework. In *Artificial Intelligence Review*, 53(2).989-1037. https://doi.org/10.1007/s10462-019-09685-9 [CrossRef]
5. Neilson, A., Indratmo, Daniel, B., & Tjandra, S. (2019). Systematic Review of the Literature on Big Data in the Transportation Domain: Concepts and Applications. *Big Data Research*, *17*, 35–44. https://doi.org/10.1016/j.bdr.2019.03.001 [CrossRef]
6. Dai, H. N., Wang, H., Xu, G., Wan, J., & Imran, M. (2020). Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies. *Enterprise Information Systems*, *14*(9–10), 1279–1303. https://doi.org/10.1080/17517575.2019.1633689 [CrossRef]
7. Rohini, P., Tripathi, S., Preeti, C. M., Renuka, A., Gonzales, J. L. A., & Gangodkar, D. (2022). A study on the adoption of Wireless Communication in Big Data Analytics Using Neural Networks and Deep Learning. *2022 2nd International Conference on Advanced Computing and Innovative Technologies in Engineering, ICACITE 2022*, 1071–1076. https://doi.org/10.1109/ICACITE53722.2022.9823439 [CrossRef]
8. Munawar, H. S., Qayyum, S., Ullah, F., & Sepasgozar, S. (2020). Big data and its applications in smart real estate and the disaster management life cycle: A systematic analysis. *Big Data and Cognitive Computing*, *4*(2), 1–53. https://doi.org/10.3390/bdcc4020004 [CrossRef]
9. Bhattarai, B. P., Paudyal, S., Luo, Y., Mohanpurkar, M., Cheung, K., Tonkoski, R., Hovsepian, R., Myers, K. S., Zhang, R., Zhao, P., Manic, M., Zhang, S., & Zhang, X. (2019). Big data analytics in smart grids: State-of-the-art, challenges, opportunities, and future directions. *IET Smart Grid*, *2*(2), 141–154. https://doi.org/10.1049/iet-stg.2018.0261 [CrossRef]
10. Arooj, A., Farooq, M. S., Akram, A., Iqbal, R., Sharma, A., & Dhiman, G. (2022). Big Data Processing and Analysis in the Internet of Vehicles: Architecture, Taxonomy, and Open Research Challenges. In *Archives of Computational Methods in Engineering* (Vol. 29, Issue 2). Springer Netherlands. https://doi.org/10.1007/s11831-021-09590-x [CrossRef]
11. Jindal, A., Kumar, N., & Singh, M. (2020). A unified significant data acquisition, storage, and analytics framework for demand response management in smart cities. *Future Generation Computer Systems*, *108*, 921–934. https://doi.org/10.1016/j.future.2018.02.039 [CrossRef]
12. Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. *Future Generation Computer Systems*, *91*, 620–633. https://doi.org/10.1016/j.future.2018.06.046 [CrossRef]
13. Mavrogiorgou, A., Kiourtis, A., Perakis, K., Miltiadou, D., Pitsios, S., & Kyriazis, D. (2019). Analyzing data and data sources towards a unified approach for ensuring quality of end-to-end data and data sources in healthcare 4.0. *Computer Methods and Programs in Biomedicine*, *181*, 1-10. https://doi.org/10.1016/j.cmpb.2019.06.026 [CrossRef]
14. Geng, D., Zhang, C., Xia, C., Xia, X., Liu, Q., & Fu, X. (2019). Significant data-based improvements to data acquisition and storage systems for designing industrial data platforms. *IEEE Access*, *7*, 44574–44582. https://doi.org/10.1109/ACCESS.2019.2909060 [CrossRef]
15. Jimenez-Marquez, J. L., Gonzalez-Carrasco, I., Lopez-Cuadrado, J. L., & Ruiz-Mezcua, B. (2019). Towards a big data framework for analysing social media content. *International Journal of Information Management*, *44*, 1–12. https://doi.org/10.1016/j.ijinfomgt.2018.09.003 [CrossRef]
16. Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, K., Martinez-Hernandez, V., Sanchez, V., & Perez-Meana, H. (2018). A Web Scraping Methodology for Bypassing Twitter API Restrictions.*Sentiment Analysis*, 1–7. http://arxiv.org/abs/1803.09875
17. Kaur, D., Aujla, G. S., Kumar, N., Zomaya, A. Y., Perera, C., & Ranjan, R. (2018). Tensor-Based Big Data Management Scheme for Dimensionality Reduction Problem in Smart Grid Systems: SDN Perspective. *IEEE Transactions on Knowledge and Data Engineering*, *30*(10), 1985–1998. https://doi.org/10.1109/TKDE.2018.2809747 [CrossRef]
18. Chen, D., Chen, Y., Brownlow, B. N., Kanjamala, P. P., Arredondo, C. A. G., Radspinner, B. L., & Raveling, M. A. (2017). Real-time or near real-time persisting daily healthcare data into HDFS and an Elastic Search index inside a big data platform. *IEEE Transactions on Industrial Informatics*, *13*(2), 595–606. https://doi.org/10.1109/TII.2016.2645606 [CrossRef]
19. Faheem, M., Butt, R. A., Ali, R., Raza, B., Ngadi, M. A., & Gungor, V. C. (2021). CBI4.0: A cross-layer approach for big data gathering for active monitoring and maintenance in the manufacturing industry 4.0. *Journal of Industrial Information Integration*, *24*, 1-17. https://doi.org/10.1016/j.jii.2021.100236 [CrossRef]
20. Liu, M., Butt, R. A., Ali, R., Raza, B., Ngadi, M. A., & Gungor, V. C. (2021). CBI4.0: A cross-layer approach for big data gathering for active monitoring and maintenance in the manufacturing industry 4.0. *Journal of Industrial Information Integration*, *24*, 1-17. https://doi.org/10.1016/j.jii.2021.100236 [CrossRef]
21. Henry, D. (2021). TwiScraper: A Collaborative Project to Enhance Twitter Data Collection. *WSDM 2021 - Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 886–889. https://doi.org/10.1145/3437963.3441716 [CrossRef]
22. Mendhe, C. H., Henderson, N., Srivastava, G., & Mago, V. (2021). A Scalable Platform to Collect, Store, Visualize, and Analyze Big Data in Real Time. *IEEE Transactions on Computational Social Systems*, *8*(1), 260–269. https://doi.org/10.1109/TCSS.2020.2995497 [CrossRef]
23. Sharma, G., Srivastava, G., & Mago, V. (2020). A Framework for Automatic Categorization of Social Data into Medical Domains. *IEEE Transactions on Computational Social Systems*, *7*(1), 129–140. https://doi.org/10.1109/TCSS.2019.2950153 [CrossRef]
24. Shah, N., Willick, D., & Mago, V. (2022). A framework for social media data analytics using Elasticsearch and Kibana. *Wireless Networks*, *28*(3), 1179–1187. https://doi.org/10.1007/s11276-018-01896-2 [CrossRef]
25. Tavares, R. C., Carvalho, M., Câmara Júnior, E. P. M., de Britto e Silva, E., Vieira, M. A. M., Vieira, L. F. M., & Krishnamachari, B. (2019). FWB: Funnelling Wider Bandwidth algorithm for high-performance data collection in Wireless Sensor Networks. *Computer Communications*, *148*, 136–151. https://doi.org/10.1016/j.comcom.2019.09.015 [CrossRef]
26. P. Le Noac'h, A. Costan and L. Bougé, "A performance evaluation of Apache Kafka in support of big data streaming applications," 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 4803-4806, doi: 10.1109/BigData. 2017.8258548. [CrossRef]
27. Y. Roh, G. Heo and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328-1347, 1 April 2021, doi: 10.1109/TKDE.2019.2946162. [CrossRef]
28. S. Vyas, R. K. Tyagi, C. Jain and S. Sahu, "Literature Review: A Comparative Study of Real Time Streaming Technologies and Apache Kafka," 2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2021, pp. 146-153, doi: 10.1109/CCICT53244.2021.00038. [CrossRef]

AUTHORS PROFILE

Sowmya R, Research Scholar Education: pursuing Ph.D. program in Computer Science and Engineering. The publication includes "Data Mining with Big Data" – DOI: 10.1109/ISCO.2017.7855990, Analysis and Verification of Video Summarization using Shot Boundary Detection, American International Journal of Research in Science, Technology, Engineering & Mathematics, ISSN (Print): 2328-3491, ISSN (Online): 2328-3580

Dr. Suneetha K R, Professor, Education: Ph.D. in Computer Science and Engineering. The publication includes "Web Log Mining using Improved Version of Apriori Algorithm", International Journal of Computer Applications, 'Classification of Web Log Data to Identify Interested Users Using Decision Trees', International Journal of Ubiquitous and Communication Journal, "Performance evaluation of compact prediction tree algorithm for webpage prediction", IEEE International Conference on Emerging Trends in Information

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Technology and Engineering (ic-ETITE'20) VIT, Vellore, India, "Applications of Association Rule Mining Algorithms in Deep Learning", Accepted in IEEE International conference on Electronics and Sustainable Communication Systems (ICESCS 2020) Hindustan Institute of Technology, CITATIONS-344, HI INDEX-6, I10 INDEX 5