An Automated Framework for Enterprise Financial Data Pre-processing and Secure Storage

Sirisha Alamanda[1]
Research Scholar, Department of Computer Science
Jawaharlal Nehru Technological University (JNTU-H)
Hyderabad, India

Dr. Suresh Pabboju[2]
Professor, Department of Information Technology
Chaitanya Bharti Institute of Technology
Hyderabad, India

Dr. G. Narasimha[3]
Professor, Department of Computer Science
Jawaharlal Nehru Technological University (JNTU-H)
Hyderabad, India

***Abstract*—The analysis on the financial data is highly crucial and critical as the results or the conclusion communicated based on the analysis can generate a greater impact on the personal and enterprise scale business processes. The primary source of the financial data is the business process and often the data is collected by automation tools deployed at various points of the business process data flow. The data entered in the business process is primary done by the stake holders of the process and at various levels of the process the data is modified, translated and sometimes completed transverter, due to which the impurities or anomalies are introduced in the data. These impurities, such as outliers and missing values, cause a high impact on the final decision after processing these datasets. Hence an appropriate pre-processing for financial data is the demand of the research. A good number of parallel research outcomes can be observed to solve these problems. Nonetheless, majority of the solutions are either highly time complex or not accurate effectively. Thus, this work proposes an automated framework for identification and imputation of the outliers using the iterative clustering method, identification and imputation of the missing values using Differential count based binary iterations method and finally the secure data storage using regression based key generation. The proposed framework has showcased nearly 100% accuracy in detection of outliers and missing values with highly improved time complexity.**

Keywords*—Financial data pre-processing; outlier treatment; missing value treatment; regression; differential iterations; iterative clustering

I. Introduction

The financial data is primarily considered to be time series data, which is variant to the time. The major complexity with the time series data analysis is two. Firstly, the speed of data change is very high. Thus, the algorithm designed to analyse the data, must be highly time efficient. Secondly, the time series data is collected from various sources, thus, the format of the data is also highly critical. These problems are well furnished in the work by C. Chatfield et al. [1]. Nonetheless, the time series data sets are the only option for a valid data analysis for making financial decisions as these financial decisions are expected to be highly time dependent [13].

Many of the cases, it is relevant to analyse the data using neural network-based algorithms due to the factor that these algorithms are less time complex and can generate better results with moderate accuracy. The work by L. Montesdeoca et al. [2] establishes significant observations in concluding the benefits of such algorithms on the financial data. Nevertheless, the primary demand of any neural network-based algorithm is highly sanitized dataset. Thus, the pre-processing of any financial data is highly expected. Also, many of the cases, a big enterprise primarily relies on the outcomes from small business units, which demands final data aggregation for the enterprise as showcased in the work by T. Cook et al. [3]. During such aggregation operations, it is highly possible to receive the final dataset with huge impurities, majorly missing values. Henceforth, it is conclusive that, the pre-processing of the financial data [Fig. 1] is a highly expected feature in any framework. Thus, this research focuses on building a framework for data pre-processing and storage security for business-critical data [19].

A. Research Problem

For financial risk analysis, a plethora of techniques has been created. In general, traditional unsupervised methods for clustering and classification do not provide adequate accuracy and semantics, while supervised approaches for classification and clustering depend on a significant quantity of training data.

B. Motivations and Objectives

This article investigates the semi-supervised scheme for financial data prediction, in which accurate predictions may be anticipated with a bit of quantity of labeled data, as shown in the previous paper. Existing semi-supervised methods have difficulty achieving acceptable results with financial data because of a lack of significant distinguishability across variables. Rather than simply propagating the input labeled data, we transform the input labeled hints to the prior global probability and propagate the 'soft' prior probability to learn the posterior probability to enhance performance.

C. Contributions

The purpose of this paper is to present automated framework for identification and imputation of the outliers using the iterative clustering. For low-security applications that only need modest levels of protection, it is necessary to provide sufficient security Secure Storage for Financial Data. The rest of the work is furnished such as, in the Sections II and III, the fundamental concepts of the data pre-processing and encryption methods for data at rest is analyzed and understood,

in the Section IV, the parallel research outcomes are critically reviewed, in Sections V and VI, the proposed solutions are proven using mathematical models and the relevant algorithms are designed with desired framework, in Section VII, the obtained results are discussed and further in Section VIII, the results and the frameworks are compared with parallel research outcomes and finally in Section IX, the final research conclusion is presented.

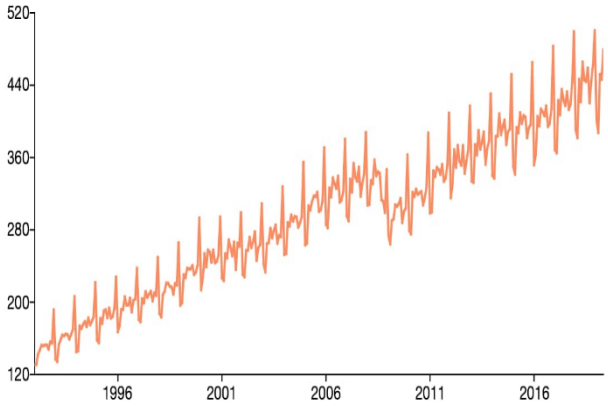

Fig. 1. Financial Data Model – Trend Based Data.

II. Data Pre-processing Fundamentals

After the primary context setting of the research outlines, in this section of the work, the fundamental strategies for data pre-processing are discussed. Due to the nature of the data collections, often the collected data is prune to the impurities which are primarily classified as outliers, missing values or sometimes noises in data analytics terms.

Thus, in this section of the work, the focus is to realize the fundamentals of these above-mentioned data defects in terms of identification and the data imputation or data cleaning.

A. Outlier Detection Fundamentals

Firstly, the outlier detection and removal process are analysed.

Assuming that the total dataset is denoted as D[] and the set of attributes forming the dataset is A[], where each and every attribute can be considered as A_i. Thus, for n number of attributes, the relation can be formulated as,

$$D[] = A[] = \langle A_1, A_2, A_3, ...A_n \rangle \tag{1}$$

Assuming that, each attribute is having the specified domain or the set of values under that attributes, which is denoted as λ. Thus, this relation can be formulated for a total of m number of observations as,

$$A_i = \langle \lambda[1], \lambda[2], ...\lambda[m] \rangle \tag{2}$$

The detection of the outlier is the process, where the values not in the range of valid or acceptable values for any specified attribute shall be identified and further eliminated or replaced with the calculated values. The generic method for identification of the valid value range can be demonstrated in two different ways as.

- Firstly, the calculation of the mean value can be performed for any specified attribute and then each value under the same attribute can be compared with the mean value to identify the deviations. The set of deviations can again be summarized as mean deviation and further be compared with each deviation obtained earlier. Finally, the deviations are above the mean deviation or standard deviation, those connected attributes can be identified as outliers.

Assuming the mean value for the domain λ is M, Thus, this can be formulated as,

$$M = \frac{\sum_{j=1}^{m} \lambda[i]}{m} \tag{3}$$

Further, the deviation set, DS[] can be calculated as,

$$DS[] \leftarrow \bigcup_{j=1}^{m} \left| \lambda[i] - M \right| \tag{4}$$

Again, the calculation of the standard deviation, σ, can be formulate as,

$$\sigma = \frac{\sum_{k=1}^{r} DS[k]}{(m)} \tag{5}$$

Here r is the length of the deviation set.

Further, the differences between the deviations from the DS[] set must be calculated and if any deviations are more than the σ, then those connected attribute values shall be identified as outliers, as:

$$Iff \left| DS[i] > \sigma, DS[i] \rightarrow A_X \right. \tag{6}$$

Then, A_X can be identified as outlier for the specified attribute domain.

- Secondly, for every attribute, based on the domain specificity or the nature of the data, a pre-decided valid value range is often assigned to the domain. Any data items violating the range, shall be considered as outliers.

From the Eq. 2, assuming that the limit or the valid value range of the specific attribute is α and β, respectively. This can be formulated as,

$$A_i = \sum_{j=1}^{m} \lambda[j], j_{\min} = \beta, j_{\max} = \alpha \tag{7}$$

Thus, any value beyond these limits must be considered as outliers.

For, either of the cases, the imputation method is fundamentally same. If any of the attributes, $\lambda[]$, are identified as outliers, then the mean deviation must be adjusted to bring to the normal value.

B. Missing Value Detection Fundamentals

Secondly, the missing values are one more type of complications in the datasets for achieving better and accurate results. The values under any specified attribute domain can be seen missing and these missing values can result into errors in the calculations on the same dataset. From Eq. 2, if any of the attribute value is null or empty, denoted as Φ, then this can be presented as,

$$\lambda[j] \rightarrow \Phi \tag{8}$$

Thus, $\lambda[j]$ can be identified as missing value and must be imputed in order to pre-process the dataset.

The imputation method for the missing value is fundamentally relies on moving average method for calculating the replacement values. Mathematically, if the $\lambda[j]$, which is j^{th} element in the dataset, then the moving average, Υ, till the $(j-1)^{th}$ element must be calculated as,

$$\Upsilon = \frac{\Upsilon + \sum_{i=1}^{j-1} \lambda[i]}{\langle \lambda[1], \lambda[2], \lambda[3], ..., \lambda[i-1] \rangle / i} \tag{9}$$

Once, the moving average is calculated, then the imputation process can be realized as,

$$\lambda[j] \leftarrow \Upsilon \tag{10}$$

C. Noise Detection Fundamentals

Finally, the last type of impurity in the dataset can be identified as noises. It is important to realize that, each data, whether represented in textual format or other formats, must be collected and stored using various devices which are often magnetically operated and during transmission it is possible to have a small amplitude shift in the data signal. Most of the times, the amplitude shift is very minor and can be ignored. However, in case of the financial data, the minor shift in the data signal can result into few decimal points in the data values, which can certainly change the course of the processing and decision-making process.

The fundamental process of detection of the noise in the data is to identify the standard length of the data values and if the data values are identified with added or extra length than the regular or pre-defined, then the extra length of the data must be reduced computationally to remove the noise from the data.

The details of the imputation process are discussed in the further sections of this work.

Henceforth, after the detailed discussion on the data pre-processing fundamentals, which is highly important to identify the gaps in the present research and further helpful in formulating the research problems, in the next section of this work, the secure storage options are identified.

III. Storage Security Fundamentals

The data security for any domain of computing is the integral part of the framework and the need for storage security cannot be ignored. Thus, in this section of the work, the fundamentals of storage security aspects are analyzed mathematically.

The primary concept of the data security at rest or the storage security relies on the concept of encryption and decryption.

From Eq. 2, assuming that any single data item under any specified domain can be considered as $\lambda[x]$. This can be presented as,

$$\lambda[x] \subseteq A[] \subseteq D \tag{11}$$

From the fundamental concepts of the encryption methods, with the two randomly selected prime numbers, as X and Y, and with the random hash e, the encryption key, KP, can be formulated as,

$$KP = \{(X.Y), e\} \tag{12}$$

And the decryption key, KPP, can be formulated as,

$$KPP = e^{-1} \% \left| (X-1).(Y-1) \right| \tag{13}$$

Thus, before storing the data item, $\lambda[x]$, the encryption process is established and the new encrypted data, $\lambda'[x]$ is generated as,

$$\lambda'[x] = \lambda[x]^e \% (KP) \tag{14}$$

Further, during the retrieval process, the decryption process must be invoked as,

$$\lambda[x] = \lambda'[x]^{KPP} \% KP \tag{15}$$

Nevertheless, during a financial data processing, the regular encryption and decryption process must be highly time complex and can deteriorate the complete process. Hence, the regular encryption-decryption process is not suitable for financial data storage security. The solution to this problem is formulated in the upcoming sections of this work.

Further, in the next section of the work, the parallel research outcomes on the data pre-processing and security are discussed.

IV. Related Work

After the detailed understanding of the fundamental strategies of the pre-processing techniques and data security at rest, in this section of the work, the parallel research outcomes are analyzed.

The primary method for analyzing the financial data for detection of anomalies or detection of the trends is the method of clustering as discussed in the work by A. Pazienza et al.[4]. However, the generic method of clustering can be sufficient for detection of the groups based on properties, but the detection of outliers or any other anomalies can be highly difficult by the generic clustering methods. Henceforth, this method is criticised by the research community. This method can be further enhanced with the method proposed in the work by S. Squires et al. [5], where the metric ranking system can be deployed to speed-up the process of clustering [12]. Nonetheless, the outcome of such ranking methods will rank the attributes and further, can reduce the number of attributes from the actual datasets. During this process, the attributes with higher number of anomalies can be directed removed by such methods, which leads to information loss [14].

In the other hand, the work formulated by Y. Kawachi et al. [6] demonstrates the use of encoding method, where the actual data items are encoded with differential parameters to identity the anomalies such as missing values or outliers [15]. This existing method is highly effective in terms of identification of anomalies, but due to higher time complexity, the applicability for financial data pre-processing is arguable. Nevertheless, the same approach is widely adopted in managing web application data as showcased by H. Xu et al. [7]. Many of the parallel research attempts have tried to reduce the time complexity of this existing method by introducing the benefits of machine learning driven optimization. One such work is worth the mention is the work by S. Squires et al. [8]. The bottleneck with such solutions is the definitive time complexity and limitations of managing a large search space. Hence, once again this method is also arguable for applicability on financial data [17].

Yet another direction of the research is proposed by A. Prügel-Bennett et al. [9] by combining the features of the encoding method with non-attribute reduction methods. These types of solutions can be applied on the financial datasets conditionally based on the model complexity limits.

Further, in the recent time, a good number of tools can be identified to automate the process of anomaly detection and reduction for the datasets as listed by A. Paszke et al. [10]. Nonetheless, these tools are not specialized for financial data managements.

Henceforth, with the detailed understanding of the parallel research outcomes, in the next section of the work, the identification of the problems and the proposed solutions are furnished.

V. Problem Identifications and Proposed Solutions

After the detailed understanding of the pre-processing fundamentals, storage level security and the outcomes from the parallel researches, in this section of the work, the proposed solutions to the identified problems are discussed.

In the due course of study, this work has identified the following issues in the current parallel research outcomes as,

- The detections of the outliers are primarily focused on the mean deviation, which is a time complex process for higher number of elements in any datasets.
- Secondly, the missing value identification process is also highly vulnerable due to the fact that, any dataset can represent the missing values in any format as #NA or NA or "?" or blank. Thus, the formal process of identification of the missing values must be formulized to detect any of the specified notations. Also, for a large dataset, detection of the missing values after verification of individual elements can be highly time complex. Thus, the time efficient solution must be designed.
- Thirdly, during the analysis of the encryption process for data at rest, the additional time required for encryption and the decryption process is slowing down the process for data analysis, which must be reduced. Thus, a time efficient encryption solution, specified for financial data, must be designed [16].

Further, the proposed solutions are discussed with the mathematical models.

Lemma – I: The time complexity for detection of the outliers can be significantly reduced with the help of proposed deep clustering method.

Proof: The most enhanced method for grouping a set of elements or forming set of groups based on specified (supervised) or un-specified (un-supervised) set of characteristics is clustering. This principle is applied in the proposed method for identification of the outliers.

Firstly, from the Eq. 1 and 2, the total number of attributes are n and each attribute domain have a length of m. Thus, the total number of elements, k, can be formulated as,

$$k \leftarrow n.m \tag{16}$$

Also, from the Eq. 9, it is significant to realize that, for calculating the deviations or difference among m number of elements, a total of (m-1) number of comparisons are needed and for calculating the final standard deviation, one additional step to be performed. Hence, the total time complexity, t_1, to identify the outlier can be formulated as,

$$t_1 = n.(m-1)+1 = O(n^2),\ n \approx m \tag{17}$$

Now, focusing on the proposed method for identification of the outlier, the following formulations can be designed.

Assuming that the complete dataset is denoted as D [] and each attribute in the dataset is assumed to be presented as, A_x for total of n number of attributes. Hence, the following relation can be formed.

$$D[] \rightarrow < A_1, A_2, A_3, ...A_n > \tag{18}$$

Here, each and every attribute is considered to have their own domain with m number of values each and the data elements are denoted as D_i, which can be represented as,

$$A_x[] = \sum_{i=1}^{m} D_i \tag{19}$$

Further, the Euclidian distance between the data points can be considered as the similarity measure and the total distance set is represented as $\lambda[]$, then,

$$\lambda[] = \int_{i=1}^{m-k} \left| D_i - D_{i+1} \right| \tag{20}$$

Further, the Euclidian distance between the elements of $\lambda[]$ are calculated,

$$\bar{\lambda}[] = \int_{i=1}^{m-k-1} \left| \lambda_i - \lambda_{i+1} \right| \tag{21}$$

The new $\bar{\lambda}[]$ set defines the relation between the elements based on their similarities.

Furthermore, the repetitive iteration of the Eq. 20 can measure the similarities with deeper and contextual aspect, which can be represented as,

$$\bar{\lambda}_k[] = \int_{i=1}^{m-k} \left| \bar{\lambda}_i - \bar{\lambda}_{i+1} \right| \tag{22}$$

Thus, based on the similarity measures of Euclidian distance of the elements and the final cluster centroids can be calculated as,

$$C[] = \bar{\bar{\lambda}}_k[] = \frac{\bar{\lambda}_k[]}{\left| \lambda_i - \lambda_{i+1} \right|_{i=0}^{n}} \tag{23}$$

Here, the primary factor to be considered as in every iteration as discussed in the Eq. 20 will continuously reduce the number of data items to be analysed for outlier detections. Hence, the time complexity, t_2, can be formulated as:

$$t_2 = \frac{n.m}{2} + \frac{n.m}{4} + \frac{n.m}{8} + \frac{n.m}{16} = O(\log_2 n.m) \tag{24}$$

Naturally, the proposed method showcases a much lower time complexity, than the regular method and $t_2 << t_1$.

Further, the missing value detection problem is analyzed here.

Lemma 2: The detection of the missing values, using the proposed domain count iterative method, reduces the time complexity.

Proof: The domain count of any dataset shall be realized as the maximum number of elements without the missing or null values. Hence, the maximum count will ensure that the maximum number of elements are considered without the missing values and in case of all missing values, the complete tuple is ignored.

Assuming that, the total dataset, DS[], is a collection of multiple domains, D[], and each domain is again collection of multiple data points, D_i. Thus, for a n number of domains or attributes, the initial relation can be formulated as,

$$DS[] = \langle D[](1), D[](2), D[](3)....D[](n) \rangle \tag{25}$$

Also, assuming that each domain is consisting of m number of data points, thus, this relation can be formulated as,

$$D[](i) = \langle D_1, D_2, D_3, ..., D_m \rangle \tag{26}$$

Further, assuming that, the method Φ, is responsible for identification of the number of data points without the missing or null values. Then, $\hat{\lambda}$ being the count of data points, this proposed function can be formulated as,

$$\hat{\lambda} = \Phi(D[](i)) \tag{27}$$

Subsequently, the count of data points from each domain can be presented as $\hat{\lambda}[]$ and can be formulated as,

$$\hat{\lambda}[] = \forall \left(D[](X) \right) \tag{28}$$

Further, assuming the maximum value from the $\hat{\lambda}[]$ collection is δ, and then this can be formulated as,

$$\delta = {}_{MAX} \left(\hat{\lambda}[] \right) \tag{29}$$

Further for domain the count of the number of data points, Y, must be compared with the maximum data point count, X, using the divide and conquer method as following:

$$\begin{cases} Iff\ \ \delta > \hat{\lambda}[i], \\ Then, Compare\ \delta/2 > \prod_{j=1}^{i/2} (\hat{\lambda}[j]) \\ Else, Compare\ \delta/2 > \prod_{j=i/2+1}^{i} (\hat{\lambda}[j]) \end{cases} \tag{30}$$

Henceforth, if the count of data points is less than the expected count of the data points in first or second half of the domain, then the process must be repeated to identify the missing values only in that half of the domain and the process shall be repeated iteratively to identify all missing values.

Further, the time complexity of this proposed method is analysed against the generic method.

Assuming that, a total of k number of iterations has to be performed for n number of domains, thus the time complexity, T_1, can be formulated as,

$$T_1 = 1 + n/2 + n/4 + ... n/k \tag{31}$$

This can be re-written as,

$$T_1 = O(k \log_2 n) \tag{32}$$

In the other hand, for the similar identification, using the generic methods, thus the time complexity, T_2, can be formulated as,

$$T_2 = k * n \tag{33}$$

It is natural to realize that

$$T_1 << T_2 \tag{34}$$

Hence, the proposed method for outlier detection significantly reduces the time complexity with higher accuracy.

Finally, the solution for the encryption problem is furnished here.

Lemma 3: Generation of the security keys using regression method can be highly time efficient.

Proof: The regression method is primarily used for predictive analysis. Nevertheless, the highly time efficient characteristics of regression method can be utilized to produce random numbers. This solution focuses on the same aspect.

From Eq. 11, the data items to be encrypted are $\lambda[x]$ with the encryption key. The components of the encryption keys can be generated using the regression methods.

The proposed method starts with the selection of the random value, p(t) and the second component q(t) can be generated using the regression method as following with the regression coefficient $\beta_{pq}(t)$,

$$q(t) = \beta_{pq}(t-1).p(t) \tag{35}$$

Here, the regression coefficient can be formulated as,

$$\beta_{pq}(t-1) = \frac{q(t-2)}{p(t-2)} \tag{36}$$

Further, the hash, e, the following formulation can be utilized,

$$e(t) = \beta_{pe}(t).p(t) + \beta_{qe}(t).q(t) \tag{37}$$

And, the regression coefficients can be formulated as,

$$\beta_{pe}(t) = \frac{e(t-1)}{p(t-1)} \tag{38}$$

And,

$$\beta_{qe}(t) = \frac{e(t-1)}{q(t-1)} \tag{39}$$

Further, from the Eq. 12 to Eq. 15 can be replicated with the newly generated components for encryption and decryption.

It is natural to realize that, the time for selection of random numbers as components of the encryption methods, is highly reduced as in the proposed method only one random number must be selected and for only one number the randomness shall be verified. Thus, the overall time complexity reduced to a greater extend.

Further, in the next section of this work, based on the proposed mathematical models, algorithms are designed with the proposed framework.

VI. Proposed Algorithm and Framework

After the detailed discussion on the fundamental processes for pre-processing, storage security, problem identification and proposed mathematical solutions, in this section of the work, the proposed algorithms and the proposed framework is furnished.

Firstly, the outlier removal algorithm is furnished.

Algorithm - I: Iterative Clustering for Outlier Detection & Imputation (**ICOD-DI**) Algorithm

Input:

The initial dataset as DS[]

Output:

The outlier reduced dataset as DS1[]

Process:

Step - 1. Load the initial dataset as DS[]

Step - 2. For each element in the DS[] as DS[i]

a. List the number of attributes as A[]

b. For each attribute as A[j]

i. List the data items as I[] and for each data item I[k]

1. Cluster the I[] using K-Means Method and store in cluster set C[]
2. Calculate the cluster centroids as C1[] and further cluster the centroids and store in cluster set C2[]
3. Identify the outliers from the second cluster set C2[] as O[]
4. If O[n] is outlier,
5. Then,
6. Check for C1[k] as outlier
   a. If C1[k] is outlier,
   b. Then,
   c. Mark I[k] as outlier

ii. Calculate the moving average till I[k] as AVG[k] for O[n]

iii. Update the values O[n] = AVG[k]

c. Else,

d. DS1[i] = DS[i]

Step - 3. Return the outlier reduced dataset as DS1[]

Oddities, or exceptions, can be a difficult issue when preparing AI calculations or applying measurable strategies. They are regularly the aftereffect of mistakes in estimations or remarkable framework conditions and in this manner don't portray the normal working of the basic framework. In reality, the best practice is to carry out an exception evacuation stage prior to continuing with additional examination.

Sometimes, exceptions can give us data about confined inconsistencies in the entire framework; so, the identification of anomalies is an important interaction on account of the extra data they can give about your dataset.

Secondly, the algorithm for missing value detection is furnished here.

Algorithm - II: Differential Count Based Missing Value Detection & Imputation (**DCB-MVDI**) Algorithm

Input:

Outlier Removed Dataset, DS1[]

Output:

Missing Value reduced dataset, DS2[]

Process:

Step - 1. Load the dataset DS1[]

Step - 2. For each element in DS1[] as DS1[i]

a. List the number of attributes as A[]

b. For each element in A[] as A[i]

i. Calculate the number of elements as Count(A[i]) => C[j]

c. If C[j] > C[j+1],

d. Then,

i. Divide A[i+1] as A1[] and A2[]

ii. Calculate the number of elements as Count(A1[]) = C1 and Count(A2[]) = C2

iii. If C1 > C2,

iv. Then,

v. Divide A2[] as A11[] and A12[] and repeat the process until C1 = C2.

vi. Mark the missing values as M[] //Using Eq. 6

vii. Calculate the moving average till A[i] as AVG[k] for M[k]

viii. Update the values M[k] = AVG[k]

e. Else,

f. DS2[i] = DS1[i]

Step - 3. Return the missing value reduced dataset as DS2[]

Missing information can happen due to nonresponse: no data is accommodated at least one thing or for an entire unit ("subject"). A few things are bound to produce a nonresponse than others: for instance, things about private subjects like pay. Weakening is a sort of missingness that can happen in longitudinal investigations—for example considering advancement where an estimation is rehashed after a specific timeframe. Missingness happens when members drop out before the test closures and at least one estimation are absent.

Thirdly, the algorithm for the data storage security is discussed here.

Algorithm - III: Polynomial Regression-based Encryption and Decryption (**PRR-ED**) Algorithm.

Input:

Missing value reduced dataset DS2[]

Output:

Encrypted dataset as DS3[]

Process:

Step - 1. Load the dataset DS2[]

Step - 2. For each element in DS2[] as DS2[i]

a. List the number of attributes as A[]

b. Select the random number p(t) and calculate q(t) & e(t) //Using Eq. 35 to 39

c. Calculate the Public and Private key //Using Eq. 12 to 15

d. For each element in A[] as A[i]

i. List the data items as I[] and for each data item I[k]

ii. Encryption:

1. II[k] = I[k]%PublicKey

2. DS3[j] = II[k]

iii. Decryption:

1. DS2[j] = POW(DS3[j],PublicKey)%PrivateKey

iv. Regression:

1. p(t+1) = p(t)

2. q(t+1) = {q(t-1)/p(t-1)}.p(t)

3. e(t+1) = {e(t-1)/p(t-1)}.p(t) + {e(t-1)/q(t-1)}.q(t)

v. Repeat the process till m = length of A[i]

Relapse investigation is principally utilized for two theoretically particular purposes. In the first place, relapse examination is generally utilized for expectation and anticipating, where its utilization has significant cover with the

field of AI. Second, in certain circumstances relapse investigation can be utilized to construe causal connections between the autonomous and ward factors. Critically, relapses without anyone else just uncover connections between a reliant variable and an assortment of autonomous factors in a fixed dataset. To utilize relapses for forecast or to deduce causal connections, separately, an analyst should cautiously legitimize why existing connections have prescient force for another specific situation or why a connection between two factors has a causal understanding. The last is particularly significant when specialists desire to appraise causal connections utilizing observational information [18].

Finally, the proposed framework is furnished here [Fig. 2].

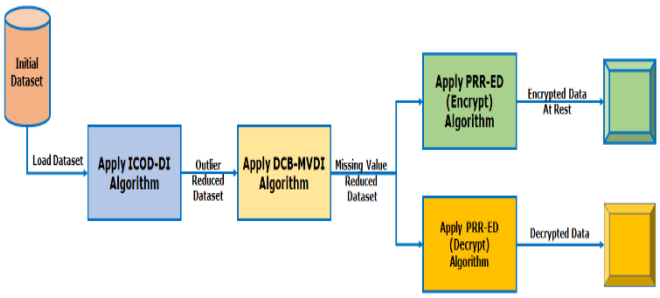

Fig. 2. Proposed Data Pre-Processing and Security Framework.

Further, in the next section of this work, the obtained results from the proposed framework is elaborated and discussed.

VII. Results and Discussions

After the detailed discussions on the problem formulation and solutions with the proposed algorithms and framework, in this section of the work, the obtained results are furnished and discussed.

The results are highly satisfactory and furnished in four phases for detailed understanding.

A. Dataset Information

Firstly, the description about the dataset [11] is furnished here [Table I].

The initial dataset conditions are also visually identified here [Fig. 3].

Here the dataset initially showcases a lot of anomalies in the form of outliers and missing values. In the next part of the results, the outlier treatment outcomes are listed.

TABLE I. Dataset Details

Table Head	Table Column Head
Number of Attributes	8
Number of Instances	2545
Number of Missing Values	2360
Number of Outliers	728

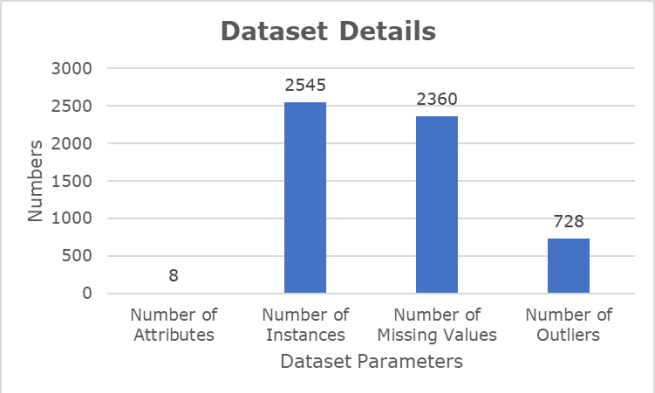

Fig. 3. Initial Dataset.

B. Outlier Treatment Outcomes

Secondly, the outlier detection results are analyzed here [Table II].

TABLE II. Outlier Treatment – Pre-Condition

Parameter Name	Type	Number of Outliers
LTV	Numeric	0
Recovery_rate	Numeric	0
lgd_time	Numeric	0
y_logistic	Numeric	0
lnrr	Numeric	728
Y_probit	Numeric	0
type	Numeric	0
class	Nominal	0

Here is it natural to realize that, the outliers are observed for only one parameter and after applying the proposed algorithm, the following outcomes are received [Table III].

TABLE III. Outlier Treatment – Post-Condition

Parameter Name	Parameter Values
Number of Outliers Present	728
Number of Outliers Detected	728
Number of Outliers Imputed	728
Accuracy (%)	100%
Time to Detect (ms)	11.02

The results are visualized graphically here [Fig. 4].

It is natural to observe that, due to the higher effectiveness of the proposed algorithm, which is proven using the mathematical model, 100% accuracy is obtained.

After the successful removal of the outliers, in order to make the data anomaly fee, in the next part of the results, the missing value treatment outcomes are discussed.

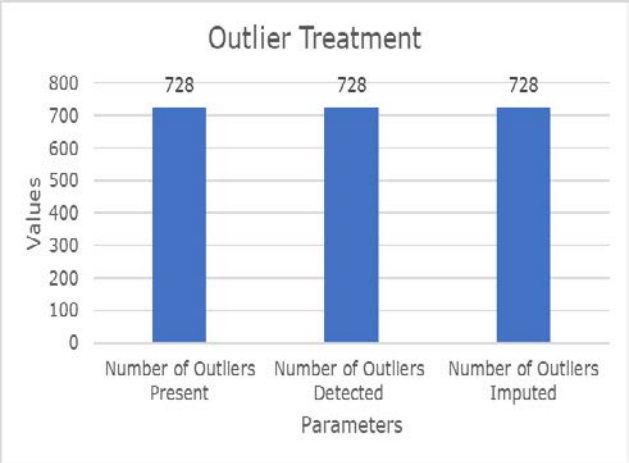

Fig. 4. Outlier Treatment.

C. Missing Value Treatment Outcomes

Thirdly, the missing values detection results are analyzed here [Table IV].

TABLE IV. MISSING VALUE TREATMENT – PRE-CONDITION

Parameter Name	Type	Number of Outliers
LTV	Numeric	0
Recovery_rate	Numeric	0
lgd_time	Numeric	0
y_logistic	Numeric	0
lnrr	Numeric	0
Y_probit	Numeric	0
type	Numeric	2360
class	Nominal	0

Here is it natural to realize that, the missing values are observed for only one parameter and after applying the proposed algorithm, the following outcomes are received [Table V].

The results are visualized graphically here [Fig. 5].

It is natural to observe that, due to the higher effectiveness of the proposed algorithm, which is proven using the mathematical model, 100% accuracy is obtained.

After the successful removal of the missing values from the dataset, in the next part of the results, the encryption and decryption at rests are discussed.

TABLE V. MISSING VALUE TREATMENT – POST-CONDITION

Parameter Name	Parameter Values
Number of Missing Values Present	2360
Number of Missing Values Detected	2360
Number of Missing Values Imputed	2360
Accuracy (%)	100%
Time to Detect (ms)	9.18

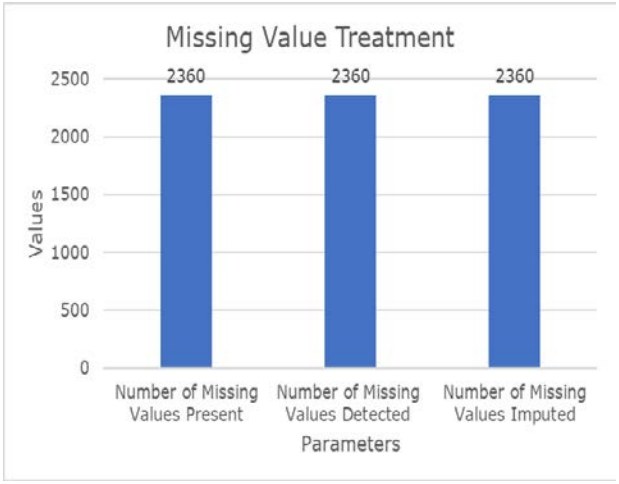

Fig. 5. Missing Value Treatment.

D. Encryption – Decryption at Rest Outcome

Finally, the encryption and decryption time complexity analysis are furnished here [Table VI]. The time complexity is tested for a total of 10 iterations here.

TABLE VI. DATA AT REST ENCRYPTION – DECRYPTION ANALYSIS

Iteration #	Key Generation Time (ms)	Encryption Time (ms)	Decryption Time (ms)
1	9	27.351	36.853
2	2	15.482	25.248
3	1	5.58	13.804
4	2	12.446	18.829
5	6	59.61	67.886
6	3	13.35	22.657
7	3	3.351	9.067
8	3	20.145	28.562
9	8	54.088	59.27
10	8	72.72	77.578

The results are visualized graphically here [Fig. 6].

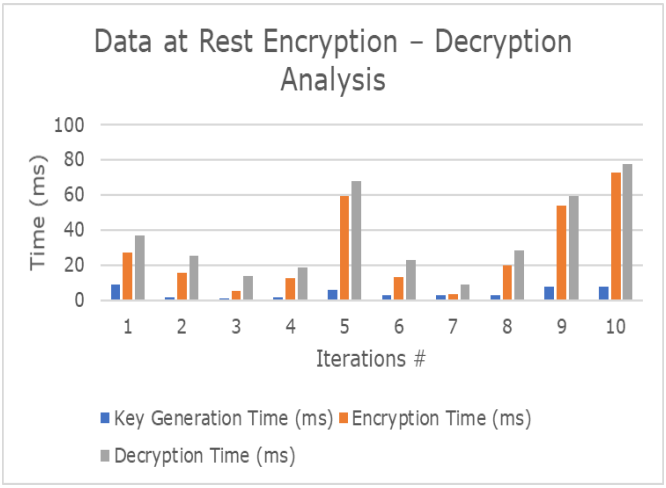

Fig. 6. Data at Rest Encryption – Decryption Analysis.

It is natural to realize that, out of 10 iterations, in 7 iterations the time complexity is reduced to a greater extend.

Further, after the detailed analysis on the obtained results, in the next section of this work, the proposed framework is compared with the parallel research outcomes.

VIII. Comparative Analysis

After the detailed analysis of the obtained results, in this section of the work, the proposed framework is compared with notable significant other parallel research outcomes [Table VII].

TABLE VII. Comparative Analysis

Author, Year	Proposed Method	Pre-Process-ing	Storage Security	Framework Complexity
A. Pazienza et al. [4], 2016	Clustering	No	No	$O(n^n)$
Y. Kawachi et al. [6], 2018	Auto Encoding	Yes	No	$O(n^3)$
S. Squires et al. [8], 2019	Variational Auto Encoder	Yes	No	$O(n^2)$
Proposed Framework, 2021	Iterative Clustering, Differential Algorithms and Regression	Yes	Yes	$O(n^2)$

Henceforth, it is natural to realize that, the proposed framework has outperformed the parallel research outcomes.

Further, in the next section of this work, the final research conclusion is presented.

IX. Conclusion

The financial data analysis is the recent trend in research and many parallel research outcomes are focusing primarily on the aspect of cleaning the data for making the dataset completely anomaly free. In the due course of study, this work identified that, firstly, the detections of the outliers are primarily focused on the mean deviation, which is a time complex process, thus needs to be optimized. Secondly, the missing value identification process is also highly vulnerable due to the fact that, any dataset can represent the missing values in any format, thus, the time efficient solution must be designed. Thirdly, during the analysis of the encryption process for data at rest, the additional time required for encryption and the decryption process is slowing down the process for data analysis, which must be reduced, thus, a time efficient encryption solution, specified for financial data, must be designed. Henceforth, this work proposes an automated framework for identification and imputation of the outliers using the iterative clustering method, identification and imputation of the missing values using Differential count based binary iterations method and finally the secure data storage using regression based key generation. The proposed framework has showcased nearly 100% accuracy in detection of outliers and missing values with highly improved time complexity for making the world of financial data analysis a better place. In future, further step can be taken to make data affinity and labelling structures more consistent by combining similarity matrices from the feature space and label space in an adaptive manner, which may be done using the label diffusion framework. It helps to make the structures of data affinity and labelling more consistent. An iterative optimization technique may be used to solve the energy function in a practical manner. Experimental findings on publicly available datasets show that the suggested approach outperforms the comparable methods in terms of overall performance.

References

[1] C. Chatfield, The analysis of time series: An introduction, Chapman and Hall/CRC, 2016.

[2] L. Montesdeoca, M. Niranjan, "Extending the feature set of a data-driven artificial neural network model of pricing financial options", 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-6, Dec 2016.

[3] T. Cook, A. S. Hall, "Macroeconomic indicator forecasting with deep neural networks", Federal Reserve Bank of Kansas City, no. 17-11, 2017.

[4] A. Pazienza, S. F. Pellegrino, S. Ferilli, F. Esposito, "Clustering underlying stock trends via non-negative matrix factorization", Proceedings of the First Workshop on Mining Data for Financial Applications, pp. 5-16, 2016.

[5] S. Squires, L. Montesdeoca, A. Pnigel-Bennett, M. Niranjan, "Non-negative matrix factorization with exogenous inputs for modelling financial data", International Conference on Neural Information Processing, pp. 873-881, 2017.

[6] Y. Kawachi, Y. Koizumi, N. Harada, "Complementary set Variational Autoencoder for supervised anomaly detection", International Conference on Acoustics Speech and Signal Processing (ICASSP). IEEE, pp. 2366-2370, 2018.

[7] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng et al., "Unsupervised anomaly detection via Variational Autoencoder for seasonal KPIs in web applications", Proceedings of the 2018 World Wide Web Conference, pp. 187-196, 2018.

[8] S. Squires, A. P. Bennett, M. Niranjan, A Variational Autoen-coder for probabilistic non-negative matrix factorisation, 2019.

[9] S. Squires, A. Prügel-Bennett, M. Niranjan, "Rank selection in nonnegative matrix factorization using minimum description length", Neural computation, vol. 29, no. 8, pp. 2164-2176, 2017.

[10] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, "Automatic differentiation in PyTorch", NIPS Autodijf Workshop, 2017.

[11] X. Zhou, Z. Pan, G. Hu, S. Tang, C. Zhao, "Stock market prediction on high-frequency data using generative adversarial nets", Mathematical Problems in Engineering, vol. 11, 2018.

[12] Yeo AC, Smith KA, Willis RJ, Brooks M (2001) Clustering technique for risk classification and prediction of claim costs in the automobile insurance industry. Intell Syst Acc Finance Manage 10(1):39–50.

[13] Kou G, Peng Y, Wang G (2014) Evaluation of clustering algorithms for financial risk analysis using MCDM methods. Inf Sci 275:1–12.

[14] Wang Y, Meng Y, Fu Z et al (2017) Towards safe semi-supervised classification: adjusted cluster assumption via clustering. Neural Process Lett 46(3):1031–1042.

[15] Ma X, Gao L, Yong X, Lidong Fu (2010) Semi-supervised clustering algorithm for community structure detection in complex networks. Physica A 389:187–197.

[16] Zhao Y, Karypis G (2001) Criterion functions for document clustering: experiments and analysis, Technical Report TR 01-40, Department of Computer Science, University of Minnesota.

[17] Tay FEH, Cao LJ (2002) ε-Descending support vector machines for financial time series forecasting. Neural Process Lett 15(2):179–195.

[18] Arratia A, Belanche LA, Fábregues L (2019) An evaluation of equity premium prediction using multiple kernel learning with fnancial features. Neural Process Lett 52:117–134.

[19] Ngoc MT, Park DC (2018) Centroid neural network with pairwise constraints for semi-supervised learning. Neural Process Lett 48(3):1721–1747.