International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I4P204

Volume 70 Issue 4, 31-43, April 2022

Original Article

An Adaptive Wolf Based Dansing System for Securing Hadoop at the Data Cleaning Stage

Saritha Gattoju[1], Vadlamani Naga Lakshmi[2]

[1]*Gitam Institute of Science, Visakhapatnam, India.*
[2]*Methodist College of Engineering and Technology, Hyderabad, India.*

[1]saritha760@gmail.com

Received: 05 January 2022 Revised: 14 March 2022 Accepted: 28 March 2022 Published: 26 April 2022

Abstract *- Nowadays a large amount of data is available for the association of authority using business decisions. Moreover, the collected data from various resources are too noisy, which affects the prediction results and accuracy. Hence, Data cleaning has been introduced to provide better data quality, but the main issues of data cleaning are time consumption and malicious attacks. In this paper, a novel Wolf based Wide Dansing System (WbWDS) is developed to provide security for data during the cleaning stage. Hence, the novel WbWDS is designed with four layers: logical, physical, execution, and data cleaning. Furthermore, wolf fitness is updated to the developed framework for enhancing the security function. In addition, the involvement of wolf fitness has afforded the finest continuous monitoring results of malicious events. Additionally, the proposed WbWDS technique is implemented in Python, and an attack is launched in the cleaning layer to check the developed method's reliability. Finally, achieved performance metrics of developed WbWDS are compared with existing methods and gained the finest results with outstanding confidential rate and low execution time.*

Keywords *- Attacks detection, Confidentiality measure, Data cleaning, Secure Hadoop application.*

1. Introduction

Data cleaning identifies and removes corrupted records from the datasets, tables, and recordsets, which includes the detection of incorrect, incomplete, irrelevant parts of data and modifying, replacing [1, 2]. Moreover, data cleaning is important as it enhances data quality and increases whole productivity [3]. All incorrect or outdated information should be removed in the data cleaning process, and the finest high-quality information must be attained [4]. Additionally, the main purpose of data cleaning is to ensure consistent, correct and usable data and clean the data by detecting errors and correcting them or preventing the error when it occurs [5, 6]. Data clearing also contains more actions such as standardising data sets, fixing spellings, correcting mistakes and syntax errors [7]. In addition, some examples of malicious data are insuring data, incomplete data, duplicate data, incorrect data, and outdated data [8]. The process of data cleaning is illustrated in fig. 1.

Combining multiple data renders many opportunities for mislabelled or duplicate data [9]. While the data is incorrect, the algorithm results are unreliable [10]. The data cleaning process contains no absolute way of predicting data because the process varies from dataset to dataset [11]. The benefit of data cleaning is that it ultimately increases all productivity and permits the highest quality information [12]. Furthermore, data cleaning is not about erasing information. It will identify how to maximise dataset accuracy without deleting information [13, 14]. It is also considered a foundational element of basic data science and plays a significant role in uncovering reliable answers and analytical processes [15]. Frequently, data cleaning helps make certain matched information easier and interact with the dataset to identify information efficiently [16]. The most common application in data cleaning is data warehouse [17]. The warehouse store contains various data from disparate sources optimised by analysing and modelling [18]. The main problem of data cleaning is high bandwidth, energy consumption, time consumption and data noise [19].

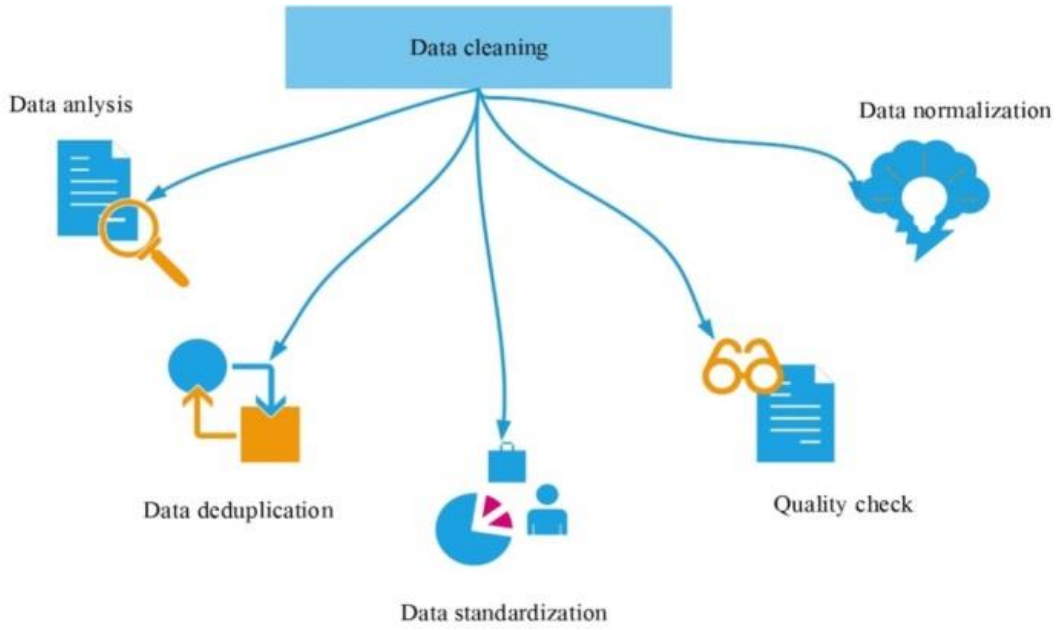

Fig. 1 Drones IoT framework

Many techniques are developed to overcome the issues of bandwidth, energy and power consumption, data noise and malicious activities [20]. Some of the existing techniques developed for data cleaning are the data cleaning method [21], open platform for data simulation [22], LOF [23] and FedClean technique [24]. However, they are still facing the problem of large power, time and energy consumption, high data noise and malicious activity. In this paper, the novel secure data cleaning technique is developed to offer sufficient privacy for the data stored in cloud storage. The main objective of this work is to provide security in the Hadoop environment while data is in the cleaning stage. In addition, the novelty of this work is designing a novel heuristic-based dansing system for the Hadoop environment to secure the data at the cleaning stage. Furthermore, the dansing system is already used in the Hadoop environment to secure the data at all stages. But, it lacks continuous monitoring to maintain the privacy range.

Grey wolf optimisation is designed in the dansing system to offer continuous monitoring. It can afford a better confidential score. The conventional models are validated and compared with the designed, optimised dansing method to know the improvement measure of the designed Hadoop security.

The key process of the designed model is discussed below,

- Initially, in the python environment, the architecture of Hadoop is modelled with all layers.
- Consequently, a novel WbWDS is designed as the security model to protect the data during the cleaning process.
- Moreover, the robustness of the proposed approach is detailed by adding some errors to the data cleaning layer.
- Finally, the parameters were calculated and validated with other existing models regarding reliability, computation time, availability, data confidentiality measure and integrity.

The arrangement of this paper is structured as follows. The related work based on Hadoop processing is detailed in section 2, and the system model and problem statement are elaborated in section 3. Also, the process of the proposed methodology is described in section 4. Finally, the achieved outcomes are mentioned in section 5, and the conclusion about the developed model is detailed in section 6.

2. Related Works

A few recent literature surveys based on data cleaning in the Hadoop environment are detailed below.

Wang et al. [21] have developed a novel data cleaning technique based on mobile edge nodes through data collection. Moreover, the angle outlier detection technique is applied to the edge node for obtaining training data using the cleaning technique. Thus the data cleaning is established by Support Vector Machine (SVM), and online learning is accepted by model optimisation. Finally, the developed technique enhanced data cleaning by maintaining data reliability, reducing energy consumption and bandwidth but with more computation time because of big data.

The classification and individuality of Internet of Things (IoT) information were studied and discussed on the Hadoop platform. Zheng and Chen [22] have proposed a

system architecture of open platforms for resource data simulations and key modules. Moreover, the data simulation technique provided a running environment, and a key technology was used for studying simulation data of IoT sensors. It predicted the accuracy and training efficiency, but the energy consumption rate was high during the process.

Xu et al. [23] have proposed a Local Outlier Factor (LOF) to enhance data quality, detect incorrect data, and data cleaning. Also, the sliding window method was introduced to split the data into different segments, with each segment containing different objects. Furthermore, the developed kernel LOF was used for calculating the degree of every segment's incorrect data. Finally, the developed method attained better performance for detecting all missing segments and is useful for big data cleaning. Hence it has obtained high bandwidth.

Generally, the increasing demand for privacy preservation largely prevents the edge node through centralised clearing. Ma et al. [24] have developed the FedClean technique for data cleaning without cooperating data privacy. Moreover, a dissimilar edge node was first generated and distributed to two servers which efficiently computed quality value frequency. Subsequently, data entities with less value were denoted as abnormal and filtered out, but the data noise rate was high compared to other techniques.

Corrales et al. [25] developed the Case-Based Reasoning (CBR) scheme for data cleaning, especially regression tasks and classification. This technique was mostly represented in the data cleaning framework, and meta featured dataset. Furthermore, the case retrieval framework was created by filter and phases; these filter retrievals minimise the number of relevant cases through the filter technique. Finally, the developed retrieval framework attained the average precision rate for the judge's ranking. The main issues of the technique were asynchronous execution and loss of data compression.

To improve data storage and reduce the execution time, Mo et al. [26] have designed the distributed file system in the Hadoop environment. Here, the data was balanced by the distributed process that tends to complete the Hadoop file sharing process in very little time. With this process, the vulnerability of malicious events has been reduced. However, it is complex in design.

Dou et al. [27] have designed the Hadoop architecture for large scale data. Moreover, the Hadoop method in large set data is vulnerable to attacks because the data present is uncontrollable. The trusted platform was implemented in the Hadoop model to improve the privacy rate, which has gained the finest outcome. But, it has taken more time to execute the process.

The mechanism of Hadoop is often utilised in big data platforms, so there is a high possibility of fault happening. Once the fault has occurred immediate process should be taken to re-order the normal process. For that, Chattaraj et al. [28] have designed a fault-tolerant mechanism in the Hadoop environment to enrich the Hadoop process. In this system, several well-known problems were balanced. But, this model required more resources to process the function.

3. System Model and Problem Definition

In big data, noisy and raw data is common since the traditional way of handling dirty data is not easily adaptable in large datasets. Moreover, raw data is incomplete, inaccurate or inconsistent because of errors in the dataset. Initially, raw datasets are updated to the system. Frequently it will detect, analyse and repair the data. Then input datasets are updated to the data cleaning process. It will clean the noisy content in the trained data, but it has issues of attack vulnerability and high time consumption to clean the data. Also, it is less reliable for cleaning data. The system model and problem definition are illustrated in Fig. 2.

The main critical task in the Hadoop framework is affording security during the data cleaning process because data cleaning may vary from dataset to dataset, so it has maximised the complexity rate. Moreover, data cleaning is most important in the Hadoop framework because of duplicate data, inaccurate data, and malicious activity. Moreover, Hadoop is the deployment paradigm widely distributed, elastic and redundant. Because of irrelevant data Hadoop framework is affected by many issues such as high time consumption, improper outcomes, malicious activity and high data noise.

It affects the security of the Hadoop application also affects digital records. This research has designed a security model in a data cleaning layer in the Hadoop framework to overcome these kinds of threats.

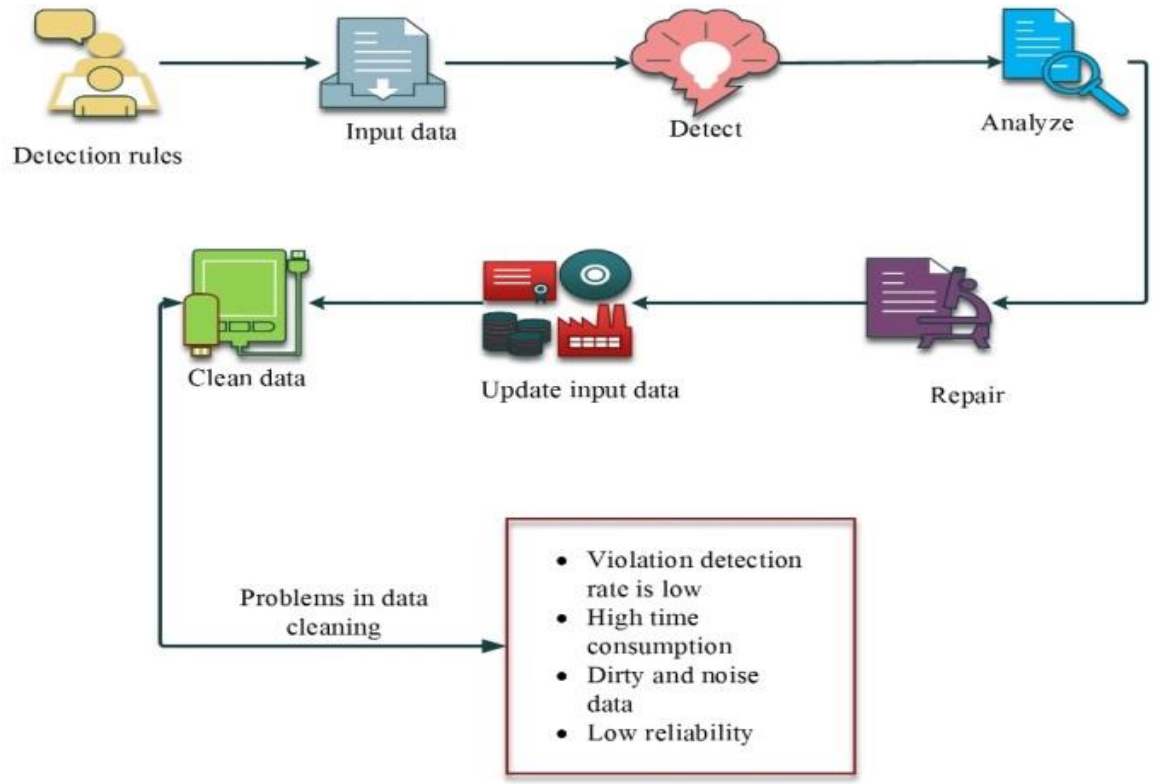

Fig. 2 Problem statement and system model

4. Proposed Methodology

The present article aims to design a novel Wolf based Wide Dansing System (WbWDS) to secure the data in the Hadoop environment in the cleaning stage. Here, enabling the wolf to function in the Hadoop environment is to maintain the integrity of the sharing data. Finally, the metrics are calculated and compared with other existing paradigms and have gained the finest outcomes. The architecture of the proposed WbWDS is detailed in fig. 3.

In addition, data cleaning is the chief process for offering better data sharing in the Hadoop architecture. So, the current research article has step-down the process in the data cleaning layer. In this proposed research, the wolf fitness improved the BiG Dansing architecture. Normally, the BiG Dansing system [33] is used for the data cleaning process in the Hadoop environment. Also, it has included three layers, namely, the execution layer, logical layer and physical layer.

4.1 Design of WbWDS in the Hadoop environment

The designed Hadoop framework contains four layers: a logical layer, a physical layer, an execution layer, and a cleaning layer. It supports the large selection of data quality rules through abstracting rule specification process. Hence, the main motive of this design is to afford the finest confidential measure during the data cleaning process. Here, the fitness of the grey wolf is used to monitor malicious events and neglect from the Hadoop framework. Furthermore, it has attained high efficiency in cleaning data through performing different kinds of physical optimisation. Thus the designed WbWDS has covered the common purpose of data processing while securing data from malicious activities during the cleaning process.

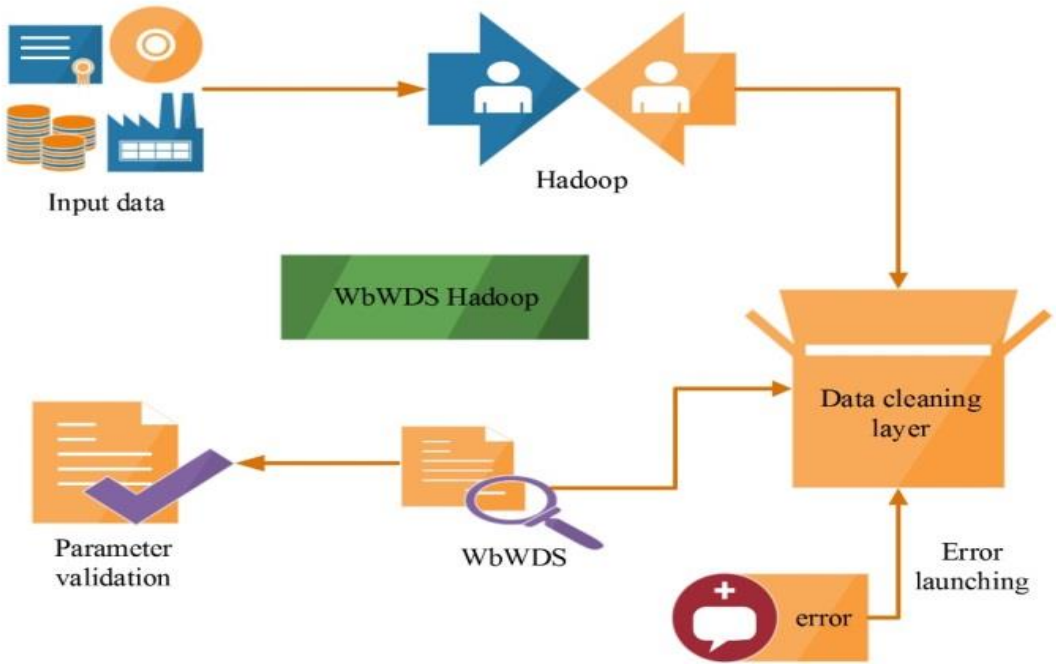

Fig. 3 Proposed model

4.1.1 Logical Layer

The logical layer enables users to define a range of data quality standards straightforwardly. It only carries the logic quality rules but does not worry about code distribution. It provides 5 logic operators, Scope, Block, Iterate, Detect and GenFix, for expressing the data quality.

Scope

It will remove the irrelevant data from the dataset, and the data unit is denoted as D, and the output of scope is a set of filtered data, or it may be an empty set.

$$Scope(D) = list(D')$$

The output of the scope allows the datasets to focus on the relevant data mainly.

Block

It will share the group of data units between the occurrences of violence through the same blocking key.

$$Block(D) = Key$$

The output of the block operator narrows the kinds of data units in arising of violation. Thus the violation arises only inside blocks and does not occur outside blocks.

Iterate

It will detail the candidate violations while the data units combine and generate candidate violations; it will operate the dataset's input list to avoid the quadratic complexity of generating candidate violations.

$$Iterate(list(D)) = D'|D_s|$$

Where D_s is denoted as the pair of datasets present in the block? It passes every unique combination of each block by producing pairs.

Detect

It will identify the list of possible violations also contains a list of violations present in the input and output dataset.

$$Detect(D'|D_s|list(D')) \rightarrow \{list(violation)\}$$

It allows the inputs and attains better performance in detecting violations.

GenFix

It generates a set of possible fixes for every given violation.

$$GenFix(violation) \rightarrow \{list\_of\_possible\_fixes\}$$

It will only assume and modify the right value also produce one possible repair in every detected violation.

4.1.2 Physical Layer

The physical layer receives information from the logical plan and transfers it to the designed physical plan. The main process of the physical layer is plan consolidation and data access. It will access the data by the specialised data access operators and joints.

4.1.3 Execution Layer

The execution layer collects the set of violations and possible fixes that can precede the cleaning process of raw input datasets. Moreover, the developed technique eliminates the violation of possible fixes by iteration, detection process.

4.1.4 Cleaning Layer

In the cleaning layer, the fitness function of Grey Wolf Optimization (GWO) is updated to protect the data. It includes four steps that are data analysis, verification, and cleaning. The design of WbWDS is illustrated in fig. 4.

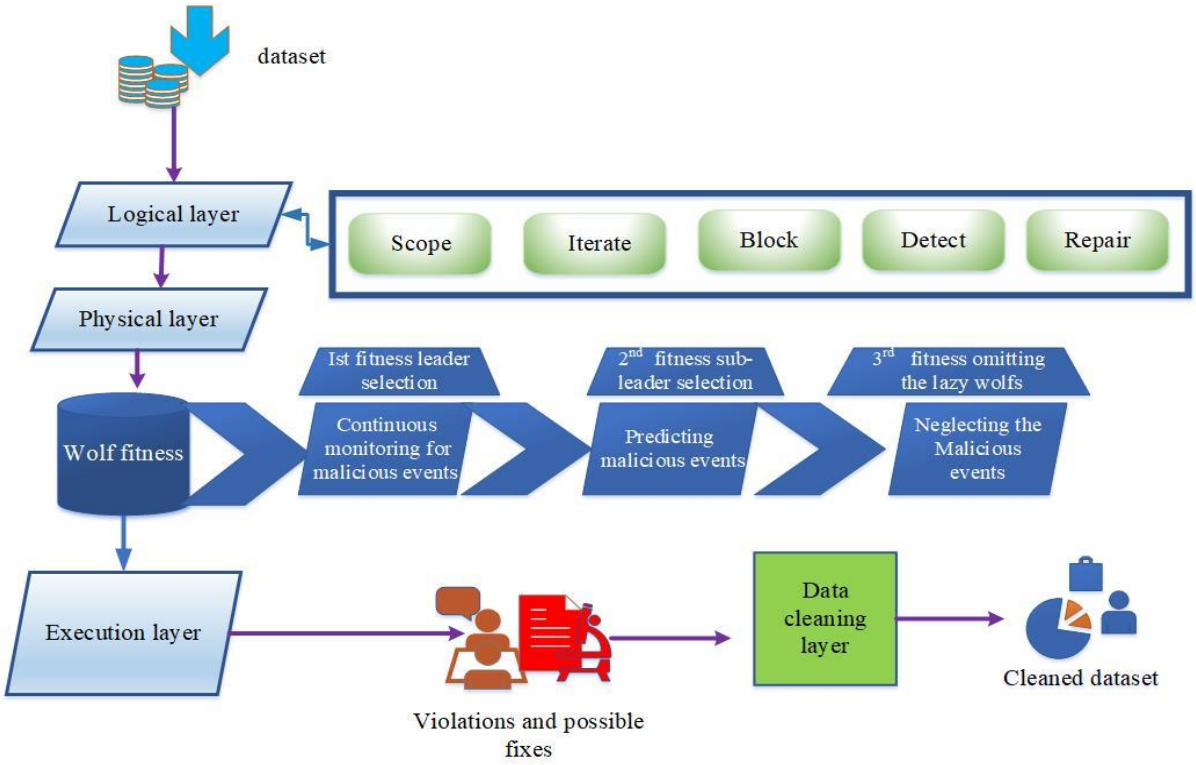

Fig. 4 Architecture of WbWDS workflow

4.2 Process of WbWDS

Initially, it analyses the data for identifying errors and inconsistencies occurring in the collected dataset. In this phase, all kinds of anomalies present inside the database are identified. Data analysis contains two processes such as data mining and data profiling. Moreover, data mining is the process of discovering specific data patterns in the dataset and data profiling is the instance emphasis of individual attributes analysis. Frequently, GWO is used for attaining the finest value outcomes, and it saves iteration time. The main aim of optimisation is to reach the prey in a short route. It includes four processes: searching for prey, encircling prey, hunting and attacking prey. To identify the position of the malicious data, alpha, beta and delta are estimated, which is the fitness of wolf functions. Moreover, the mathematical boundary of searching the position of malicious data is obtained by eqns. (1) and (2).

$$K_t = \left|P_t.SH(t) - Y(t)\right| \tag{1}$$

$$Y(t+1) = SH(t) - Q_t.K_t \tag{2}$$

where t is denoted as the current iteration. Moreover, $P_t\ Q_t$ it is considered the coefficient vector and $SH(t)$ is represented as the position of the vector of the dataset. Furthermore, $.K_t$ it is denoted as the analysis of errors present in the dataset and $Y(t)$ is considered the input dataset's position.

Then the analysed datasets enter into the verification process, which will verify the datasets based on corrupted datasets, errors, and malicious attacks. Thus the verification process is obtained using eqns. (3), (4) and (5).

$$K_t\alpha = \left|P_t\alpha.SH_\alpha(t) - Y\right| \tag{3}$$

$$K_t\beta = \left|P_t\beta.SH_\beta(t) - Y\right| \tag{4}$$

$$K_t\delta = \left|P_t\delta.SH_\delta(t) - Y\right| \tag{5}$$

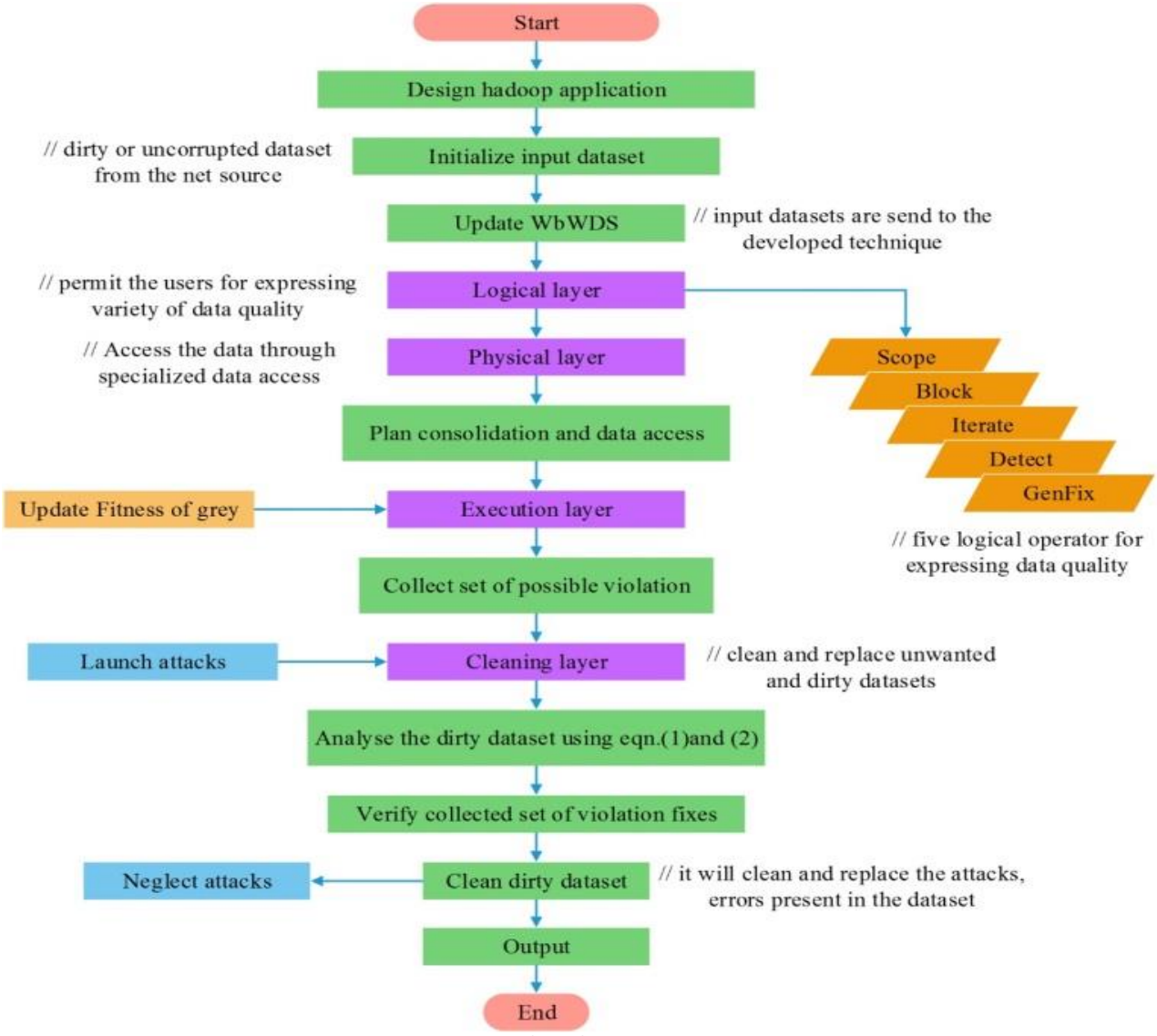

Fig. 5 Workflow of proposed WbWDS

where, $K_t\alpha$ $K_t\beta$ and $K_t\delta$ are considered as types of malicious features. It will verify the errors present in the dataset because some errors are not visible in the analysis stage. For multiple iterations, errors are visible in the verification process. Finally, the cleaning layer of the Hadoop framework is secured using eqn. (6).

$$Y(t+1)=\frac{Y_{\alpha 1}+Y_{\beta 1}+Y_{\delta 1}}{3} \tag{6}$$

Let $Y_{\alpha 1}+Y_{\beta 1}+Y_{\delta 1}$'s represent the hunting of malicious and unauthenticated events in the present dataset. The alpha, beta, and delta values are combined to predict the present malicious features in the dataset. The prediction of the malicious features present in the dataset is identified with the fitness function of GWO. Moreover, the noisy features of the raw data were successfully removed by the proposed WbWDS technique.

Moreover, for checking the reliability of the developed WbWDS technique, attacks are launched in the cleaning layer. It will detect and remove the attacks with the help of the proposed technique. Additionally, the developed technique enhances the performance of the Hadoop application. Initially, raw datasets are trained to the system that contains normal and abnormal data, and then they are updated to the developed WbWDS. This developed framework contains five layers for detecting, verifying and cleaning the data. It will identify the violation and neglect with the help of the developed framework. Moreover, the workflow of the developed WbWDS is illustrated in fig. 5.

Thus the developed framework cleans the errors, uncorrupted records, dirty data with a high confidential rate. Also, the malicious attacks present in the dataset are identified. Moreover, the identified attacks are neglected using WbWDS. It will save the execution time for cleaning the data and securing the Hadoop application.

5. Results and Discussion

The developed WbWDS is processed in Python; the success rate of the projected model is assessed by current existing mechanisms in terms of computation time, availability, data integrity and confidentiality measures. In this approach, more than 5000 datasets are utilised to validate the efficiency of the proposed model. Here, the proposed WbWDS technique identifies the malicious features neglected before the cleaning process. To check the reliability of the developed technique, attacks are launched in the cleaning layer. Hence, the developed model attained high performance in data cleaning and securing the Hadoop application.

5.1 Case Study

The data cleaning process contains identification, detection and correction of errors; also, datasets are analysed quickly. Moreover, incomplete information is generated during the data analysis process, managed during the data cleaning stage. It removes anomalies, errors, irrelevant information present in the dataset. Furthermore, it enhances the data quality, and the developed WbWDS technique is implemented for various resources such as colleges, the medical industry, and factories for securing data. The used datasets for the case study are sports, organisation, books, etc. More than 5000 datasets were used for the data cleaning process. The process of the developed framework is elaborated in fig. 6.

Initially, input datasets are updated to the developed WbWDS framework. The developed framework contains four layers: a logical, physical, execution, and cleaning layer. The logical layer identifies and collects the set of violation fixes in the dataset, and the physical layer access the data through the specialisation of data access.

Algorithm 1: Secure Data cleaning in Hadoop using WbWDS technique

Start

{

int h, c

//initialise the Hadoop parameters, here, h *represents Hadoop parameters and* c *represents cleaning stage parameters*

Enabling security $\rightarrow c$

// providing security for data at the cleaning stage

Security parameter initialisation ()

{

$data \rightarrow det\,ect(c)$

//analysing the data at the cleaning stage

$Genfix(y+1) \rightarrow c$

// fixing the hunting predictor at the cleaning layer

}

Enabling continuous monitoring ()

{

$saving \rightarrow normal \quad data \quad behaviour$

// normal data features are stored in the Hadoop cloud

$de \rightarrow c = de \rightarrow (y+1)c$

// new data entry to cleaning layer, here de is the data entry

}

Verification module()

{

If ($de = authenticated$ *)*

{

Normal authenticated data *// whether the data is authenticated or not*

}else *// this kind of data was blocked*

(

$de = \alpha, \beta, \gamma$

//malicious user)

}

stop

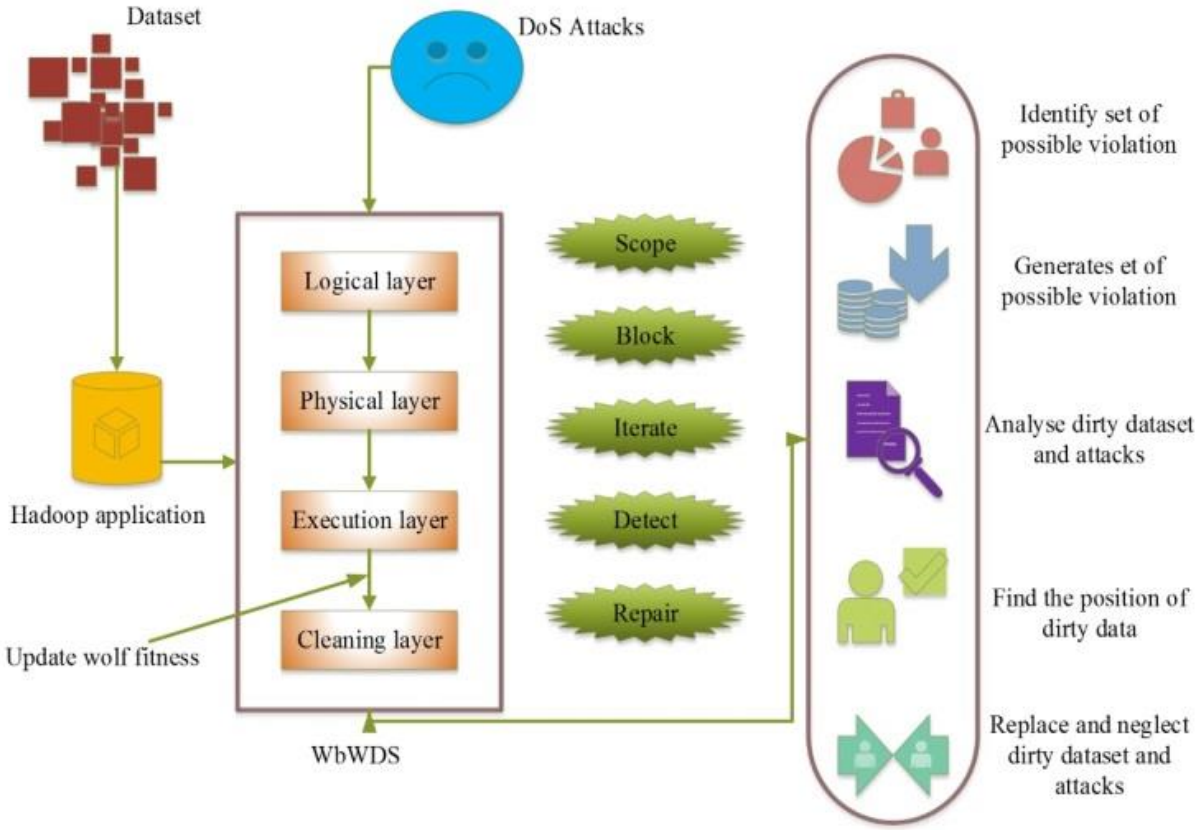

Fig. 6 Process of WbWDS technique for data cleaning

Moreover, the execution layer collects the set of violations and possible fixes using the fitness function of the wolf. The cleaning process analyses the malicious features present in the dataset. also, the position of the malicious data was found using eqns. (1) and (2). After finding the position of malicious features, the verification process begins. It will verify attacks, irrelevant data present in the dataset. Finally, verified attacks are neglected and replaced with the help of the developed WbWDS technique.

To check the reliability of the developed WbWDS technique, Denial of Service (DoS) is launched in the cleaning layer. Moreover, DoS attacks make the indented users inaccessible, accomplished through flooding targets by traffic or transferring information which triggers a collapse. Furthermore, launched DoS attacks are identified and neglected with the help of the developed WbWDS technique. Finally, it will provide high data quality and better performance in data cleaning also secure the Hadoop application.

5.2 Performance Metrics

The implementation work of the developed WbWDS is done on the Python tool, and the parameters like availability, confidentiality, integrity, and computational time were calculated. Moreover, the developed approach is validated with existing methods like Data Security in Hadoop (DSH) [26], Attacks Counter Measure in Hadoop (ACMH) [27], Big Data Hadoop Assisted (BDHA) [28], Data Integrity Verification (DIV) [29], Interconnected Distributed Framework in Cloud (IDFC) [30], Data Integrity of Cloud Computing (DI-CC) [31], Energy Analysis Based Hadoop (EABH) [32], and Big Dansing (BD) [33].

5.2.1 Computation Time (C_t)

It is the clock function from the beginning to the end of processing the code, calculated under a specific time of the execution process. The instruction count is multiplied by Cycles Per Instruction (CPI), also multiplied by clock cycle time. Moreover, CPI time is calculated based on the number of clock ticks per second and the mathematical calculation of computation time is obtained using eqn. (7).

$$C_t = I \times CPI \times C \tag{7}$$

where I is denoted as instruction count and C is represented by clock cycles per instruction? Furthermore, a comparison of computation time with other techniques is illustrated in Table 1.

Table 1. Validity of Computation Time

Techniques	Computation time (s)
DSH	0.13
ACMH	0.27
BDHA	0.32
Proposed	0.05

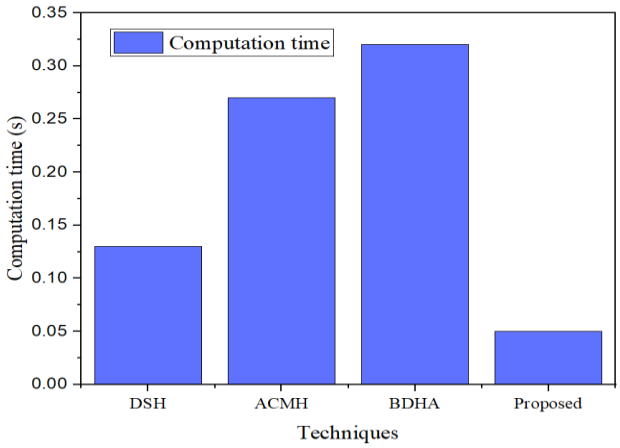

Fig. 7 Comparison of computation time

The achieved computation time is compared with other existing ACMH, BDHA, and ACMH techniques. Thus the ACMH technique and the achieved computation time for 100 tasks is 0.27s, whereas the DSH replica gained 0.13s. Moreover, the BDHS method attained a computation time of 0.32s for completing 100 tasks. The developed WbWDS technique achieved 0.05s computation time. Thus the attained computation time is very low compared to other existing techniques. Also, the comparison of computation time is detailed in fig. 7.

5.2.2 Availability

Availability is calculated by dividing scheduling working time into downtime. The system degree or subsystem is specified in a committable state which qualifies or states the availability of person or things and affordable housing. It is calculated using eqn. (8),

$$Availability = \frac{SW - DT}{SW} \tag{8}$$

Table 2. Validation of Availability

Techniques	Availability (%)
DIV	84
IDFC	78
Proposed	98.2

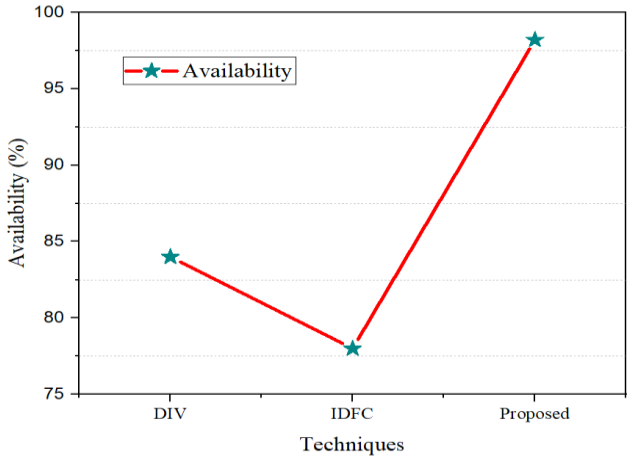

Fig. 8 Comparison of data availability

Where SW is represented as scheduling working time and DT is denoted as downtime? Moreover, validation of availability is detailed in Table 2.

The existing techniques of DIV and IDFC are compared with the developed framework. Here, DIV gained 84% availability, and the IDFC technique attained 78% availability for executing 100 tasks. While comparing other techniques developed, WbWDS attained a high data availability rate of 98.2%. Moreover, a comparison of availability is illustrated in fig. 8.

5.2.3 Confidentiality Measure

Generally, data confidentiality protects the information or data from unauthorised disclosure. Data confidentiality is the term for protecting data from unlawful, unintentional or unauthorised access. It is the privacy of information that contains share, use and authorisation. Additionally, validation of confidentiality measures with existing techniques is detailed in Table 3.

The developed WbWDS replica attained a confidentiality measure rate of 0.98% for 100 tasks. Moreover, IDFC and ACMH techniques attained 0.75% and 0.67% of data confidentiality. Furthermore, the DSH technique gained a 0.86% confidentiality measure, and DIV attained 0.80% data confidentiality. The comparison of data confidentiality measures is illustrated in fig. 9.

Table 3. Validation of Confidentiality Measure

Techniques	Confidentiality measure (%)
DIV	0.80
IDFC	0.75
ACMH	0.67
DSH	0.86
Proposed	0.98

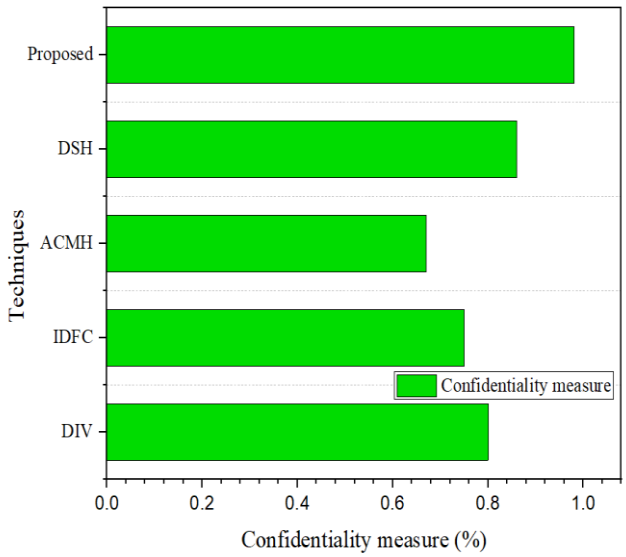

Fig. 9 Comparison of data confidentiality measure

5.2.4 Data Integrity (DI)

Data integrity measure is used to calculate the originality of the data after performing the data cleaning process. If the technique has attained a high confidential score, it has gained a high data integrity score. Thus the data integrity is measured by calculating the checksum of all trained data. Moreover, the checksum is verified with the reading and writing of data. It ensures the accuracy, completeness, and timeliness of information also prevents data tampering. The measurement of data integrity is obtained by eqn. (9).

$$DI = \frac{TP + TN}{TP + TN + FN + FN} \tag{9}$$

Where TP is represented as the true positive rate of detecting un authenticating events TN is denoted as a true negative of detecting unauthenticated data?

Moreover, it FP is denoted as the false positive rate of detecting authenticated data and FN is represented as the false-negative rate of detecting authenticated data. The comparison of data integrity is described in Table 4.

Generally, achieved data integrity is compared with other techniques such as DIV and DI-CC. Thus the DIV technique achieved data integrity for 100 tasks is 86%; the DI-CC method attained data integrity of 90% for completing 100 tasks. The developed WbWDS technique achieved 98.6% data integrity. It is very high compared to other existing techniques. Also, the comparison of data integrity is detailed in fig. 10.

Table 4. Validation of Data Integrity

Techniques	Data integrity (%)
DIV	86
DI-CC	90
Proposed	98.6

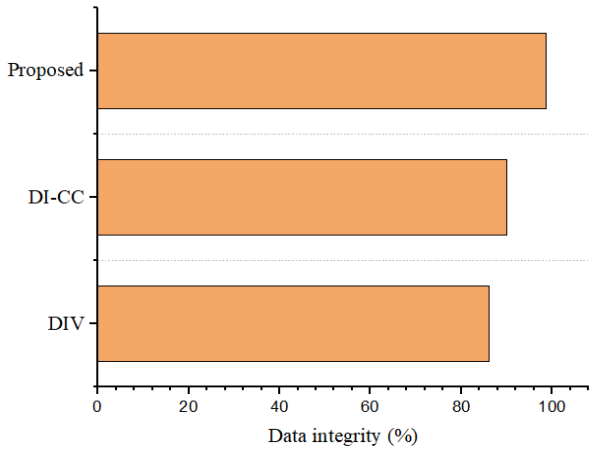

Fig. 10 Comparison of data integrity

5.2.5 Error Rate

ER is the ratio of the number of errors in the dataset to the total quantity of the transmitted dataset.

The error rate is obtained based on the data integrity measure; hence, the attained error values and their comparison are detailed in Table 5.

The existing techniques of BD and EABH are compared with the developed framework. The BD gained a 0.1% error rate, and the EABH technique attained a 0.4% error rate for executing 100 tasks. While comparing these techniques, the developed WbWDS attained a low data error rate of 0.05%. Moreover, the comparison of error rates is illustrated in fig. 11.

5.3 Discussion

Security and maintaining privacy is required for all applications to improve advancement. Hence, to improve big data security Hadoop was introduced with different layers. But still, the harmful attacks can disturb the Hadoop process in all layers. So the proposed article aims to design the security function in the Hadoop data cleaning layer. From the total outcome valuation, the developed WbWDS technique has achieved good data availability, computation time, data integrity and data confidentiality measures. The computation time and error rates are less compared to other existing techniques. Also, to know the process of the designed security model, malicious activity like error code is launched to the data cleaning layer. Hereafter, the confidential rate of the Hadoop framework is estimated.

Table 5. Validation of Error Rate

Techniques	Error rate (%)
EABH	0.4
BD	0.1
proposed	0.01

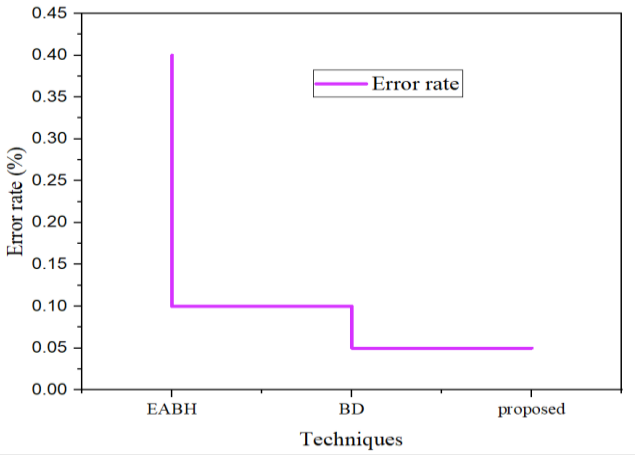

Fig. 11 Comparison of error rate

Table 6. Overall Performance Assessment

Performance assessment	
Metrics	**Performance score**
Error rate	0.01%
Data integrity	98.6%
Confidentiality rate	98%
Availability	99.2%
Computation time	0.05s

The overall performance of the presented model is tabulated in Table 6. In all metrics validation, the designed strategy has gained the best results that have proven the robustness of the proposed model.

In future, designing the security model for all Hadoop layers will improve the Hadoop performance.

6. Conclusion

Data cleaning is the key process for better prediction or other expected results in big data. So, providing security at the data cleaning stage is a crucial factor because the unstructured data contains a large amount of noise, so the cleaning process is big in the large data application. So, the present research has developed the WbWDS technique to provide the security in Hadoop environment at the data cleaning stage. Thus the developed framework contains four layers for the data cleaning process. It analyses, identifies and detects irrelevant information from the dataset also neglects the attacks.

Furthermore, the proposed WbWDS technique is implemented in a Python environment, and more than 5000 datasets were used to check the reliability of the designed security model. Additionally, the gained successive scores of the developed technique are compared with other existing techniques. The achieved result of data integrity was 98.6%, computation time 0.05s, confidentiality measure gained was 0.98% with a 0.05% error rate. The developed WbWDS technique attained a high rate of data integrity, availability, and confidentiality with fewer error rates and computation time.

References

[1] T. Barot, G. Srivastava, and V. Mago Determining Sufficient Volume of Data for Analysis with Statistical Framework, Trends in Artificial Intelligence Theory and Applications, Artificial Intelligence Practices, IEA/AIE 2020, Lecture Notes in Computer Science, Cham: Springer. 12144 (2020) 770-781.

[2] J. Lu, A. Hales, and D. Rew, Modelling of Cancer Patient Records: A Structured Approach to Data Mining and Visual Analytics, Lecture Notes in Computer Science, Cham: Springer. 10443 (2017) 30-51.

[3] A. A. Koelmans, N. H. M. Nor, E. Hermsen, M. Kooi, S. M. Mintenig, and J. D. France, Microplastics in Freshwaters and Drinking Water: Critical Review and Assessment of Data Quality, Water Res. 155 (2019) 410-422.

[4] S. Li, J. Hu, Y. Cui, and J. Hu, Deeppatent: Patent Classification with Convolutional Neural Networks and Word Embedding. 117 (2018) 721-744.

[5] F. Ridzuan, and W. M. N. W. Zainon, A Review on Data Cleansing Methods for Big Data, Procedia Comput Sci. 161 (2019) 731-738.

[6] E. A. M. Al-Masri, and Y. Bai, A Service-Oriented Approach for Assessing the Quality of Data for the Internet of Things, 2019 IEEE International Conference on Service-Oriented System Engineering (Sose). (2019) 9-97.

[7] M. Navinchandran, M. E. Sharp, M. P. Brundage, and T. B. Sexton, Discovering Critical KPI Factors from Natural Language in Maintenance Work Orders, J Intell Manuf. (2021).

[8] C. S. Wang, S. L. Lin, T. H. Chou, and B. Y. Li, An Integrated Data Analytics Process to Optimise Data Governance of the Non-Profit Organisation, Comput Hum Behav. 101 (2019) 495-505.

[9] S. Yoo, Z. Shi, B. Wen, S. J. Kho, R. Pan, H. Feng, H. Chen, A. Carlsson, P. Edén, W. Ma, M. Raymer, E. J. Maier, Z. Tezak, E. Johanson, D. Hinton, H. Rodriguez, J. Zhu, E. Boja, and B. Zhang, A Community Effort to Identify and Correct Mislabeled Samples in Proteogenomic Studies, Patterns. 2(5) (2021) 100245.

[10] H. He, W. Zhang, and S. Zhang, A Novel Ensemble Method for Credit Scoring: Adaption of Different Imbalance Ratios, Expert Syst Appl. 98 (2018) 105-117.

[11] A. Coad, And S. Srhoj, Catching Gazelles with a Lasso: Big Data Techniques for Predicting High-Growth Firms, Small Bus Econ. 55(3) (2020) 541-565.
[12] J. Miranda, P. Ponce, A. Molina, and P. Wright, Sensing, Smart and Sustainable Technologies for Agri-Food 4.0, Comput Ind. 108 (2019) 21-36.
[13] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, A Comparative Evaluation of Pre-Processing Techniques and Their Interactions for Twitter Sentiment Analysis, Expert Syst Appl. 110 (2018) 298-310.
[14] S. K. Lakshmanaprabu, K. Shankar, A. Khanna, D. Gupta, J. J. P. C. Rodrigues, P. R. Pinheiro, and V. H. C. De Albuquerque, Effective Features to Classify Big Data Using Social Internet of Things. 6 (2018) 24196-24204.
[15] A. Rizk, and A. Elragal, Data Science: Developing Theoretical Contributions in Information Systems via Text Analytics, J Big Data. 7(7) (2020) 1-26.
[16] A. Kiourtis, S. Nifakos, A. Mavrogiorgou, and D. Kyriazis, Aggregating Healthcare Data's Syntactic and Semantic Similarity Towards their Transformation to Hl7 FHIR Through Ontology Matching, Int J Med Inform. 132 (2019) 104002.
[17] Z. Li, L. Sun, and R. Higgs, Research on, and Development of, Data Extraction and Data Cleaning Technology Based on the Internet of Things, IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (Euc). (2017) 332-341.
[18] V. Palanisamy, and R. Thirunavukarasu, Implications of Big Data Analytics in Developing Healthcare Frameworks–A Review, J King Saud Univ - Comput Inf Sci. 31(4) (2019) 415-425.
[19] S. Fong, J. Li, W. Song, Y. Tian, R. K. Wong, and N. Dey, Predicting Unusual Energy Consumption Events from Smart Home Sensor Network by Data Stream Mining with Misclassified Recall, J Ambient Intell Humaniz Comput. 9 (2018) 1197–1221.
[20] S. Munawar, M. Asif, B. Kabir, A. Ullah, and N. Javaid, Electricity Theft Detection in Smart Meters Using a Hybrid Bi-Directional Gru Bi-Directional LSTM Model, Complex, Intelligent and Software Intensive Systems, Cisis 2021, Lecture Notes in Networks and Systems, Cham: Springer. 278 (2021).
[21] T. Wang, H. Ke, X. Zheng, K. Wang, A. K. Sangaiah, and A. Liu, Big Data Cleaning Based on Mobile Edge Computing in Industrial Sensor-Cloud, IEEE Trans Ind Inform. 16(2) (2020) 1321-1329.
[22] Y. Zheng, and G. Chen, Energy Analysis and Application of Data Mining Algorithms for Internet of Things Based on Hadoop Cloud Platform, IEEE Access. 7 (2019) 183195-183206.
[23] X. Xu, Y. Lei, and Z. Li, An Incorrect Data Detection Method for Big Data Cleaning of Machinery Condition Monitoring, IEEE Trans Ind Electron. 67(3) (2020) 2326-2336.
[24] L. Ma, Q. Pei, L. Zhou, H. Zhu, L. Wang, and Y. Ji, Federated Data Cleaning: Collaborative and Privacy-Preserving Data Cleaning for Edge Intelligence, IEEE Internet Things J. 8(8) (2021) 6757-6770.
[25] D. C. Corrales, A. Ledezma, And J. C. Corrales, A Case-Based Reasoning System for a Recommendation of Data Cleaning Algorithms in Classification and Regression Tasks, Appl Soft Comput. 90 (2020) 106180.
[26] Y. Mo, A Data Security Storage Method for Iot Under Hadoop Cloud Computing Platform, Int J Wirel Inf Netw. 26(3) (2019) 152-157.
[27] Z. Dou, I. Khalil, A. Khreishah, and A. Al-Fuqaha, Robust Insider Attacks Countermeasure for Hadoop: Design and Implementation, IEEE Syst J. 12(2) (2018) 1874-1885.
[28] D. Chattaraj, M. Sarma, A. K. Das, N. Kumar, Joel. J. P. C. Rodrigues, and Y. Park, Heap: An Efficient and Fault-Tolerant Authentication and Key Exchange Protocol for Hadoop-Assisted Big Data Platform, IEEE Access 6. (2018) 75342-75382.
[29] R. Saxena, and S. Dey, A Curious, Collaborative Approach for Data Integrity Verification in Cloud Computing, CSI Trans ICT. 5(4) (2017) 407-418.
[30] M. Maghsoudloo, and N. Khoshavi, Elastic Hdfs: Interconnected Distributed Architecture for Availability–Scalability Enhancement of Large-Scale Cloud Storages, J Supercomput. 76(1) (2020) 174-203.
[31] R. Saxena, and S. Dey, Data Integrity Verification: A Novel Approach for Cloud Computing, Sādhanā. 44(74) (2019) 1-12.
[32] Y. Zheng, and G. Chen, Energy Analysis and Application of Data Mining Algorithms for Internet of Things Based on Hadoop Cloud Platform, IEEE Access. 7 (2019) 183195-183206.
[33] Z. Khayyat, I. F. Ilyas, A. Jindal, S. Madden, M. Ouzzani, P. Papotti, J. A. Quiané-Ruiz, N. Tang, and S. Yin, Bigdansing: A System for Big Data Cleansing, Sigmod '15: Proceedings of the 2015 ACM Sigmod International Conference on Management of Data. (2015) 1215–1230.