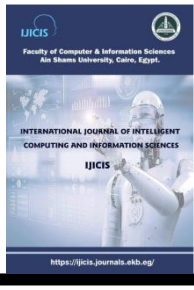
IJICIS, Vol.23, No.2, 13-27
DOI: 10.21608/ijicis.2023.188202.1249

**International Journal of Intelligent Computing and Information Sciences**

https://ijicis.journals.ekb.eg/




# DATA FUSION FOR DATA PREDICTION: AN IoT-BASED DATA PREDICTION APPROACH FOR SMART CITIES

Dina Fawzy

Department of Information Systems,
Faculty of Computer and Information Sciences, Ain Shams University
Cairo 11566, Egypt
dina.fawzy@cis.asu.edu.eg

Sherin M. Moussa*

Department of Information Systems,
Faculty of Computer and Information Sciences,
Ain Shams University,
Cairo 11566, Egypt
Laboratoire Interdisciplinaire de l'Université Française d'Egypte (UFEID Lab), Université Francaise d'Egypte, Cairo 11837, Egypt
sherinmoussa@cis.asu.edu.eg,
sherin.moussa@ufe.edu.eg

Nagwa Badr

Department of Information Systems,
Faculty of Computer and Information Sciences, Ain Shams University
Cairo 11566, Egypt
nagwabadr@cis.asu.edu.eg

Received 2023-01-18; Revised 2023-05-02; Accepted 2023-05-05

***Abstract:*** *Recently with the high implementation of numerous Internet of Things (IoT) based systems, it becomes a crucial need to have an effective data prediction approach for IoT data analysis that copes with sustainable smart city services. Nevertheless, IoT data add many data perspectives to consider, which complicate the data prediction process. This poses the urge for advanced data fusion methods that would preserve IoT data while ensuring data prediction accuracy, reliability, and robustness. Although different data prediction approaches have been presented for IoT applications, but maintaining IoT data characteristics is still a challenge. This paper presents our proposed approach the domain-independent Data Fusion for Data Prediction (DFDP) that consists of: (1) data fusion, which maintains IoT data massive size, faults, spatiotemporality, and freshness by employing a data input-data output fusion approach, and (2) data prediction, which utilizes the K-Nearest Neighbor data prediction technique on the fused data. DFDP is validated using IoT data from different smart cities datasets. The experiments examine the effective performance of DFDP that reaches 91.8% accuracy level.*

***Keywords:*** *IoT, Data Prediction, Data Fusion, KNN, Smart Cities.*

## 1. Introduction

***Corresponding Author**: Sherin M. Moussa

Information Systems Department, Faculty of Computer and Information Science, Ain Shams University, Cairo 11566, Egypt

Laboratoire Interdisciplinaire de l'Université Française d'Egypte (UFEID Lab), Université Francaise d'Egypte, Cairo 11837, Egypt

Email address: sherinmoussa@cis.asu.edu.eg, sherin.moussa@ufe.edu.eg

An metropolitan area that uses digital technology to handle its community services more effectively is referred as a "smart city" [1]. The goal of smart cities is to create a more comfortable and sustainable environment that enhances the standard of living for residents based on data prediction analysis and Internet of Things (IoT) [2]. Smart tools like sensors and actuators have been widely increased because of IoT [3]. Sensors are used to collect massive amounts of heterogeneous data from different IoT-based applications such as environmental monitoring, healthcare, smart transportation systems, IoT-based web mining and social media, IoT data streaming, and industrial plant monitoring [4]. These devices gather data from the different IoT-based applications and transmit them to platforms for processing and analysis. The problem arises from the IoT-based data generated from the connected devices with diverse digital technologies which are massive in size, highly heterogeneous, real-time, dynamic, spatial, temporal and volatile [5]. In this sense, the different types of sources, and the enormous amounts of inaccurate data collected at unusual high rates, make predicting these data inaccurate, unreliable and necessitating much processing time [6]. Thence, new data prediction processes have been prompted using different data prediction techniques to consider all these challenges [7]. Data prediction techniques such as: Naive Bayes (NB), Decision Trees (DT), K-Nearest Neighbor (KNN), regression techniques, and Neural Networks (NN) have become widely considered in IoT-based data prediction approaches [8]. Despite of the enormous data prediction processes have been presented for different IoT domains, but still they have some limitations to cope all new features of IoT data [9]. Hence, data fusion has become a popular method for maintaining and enhancing data for further data analysis [5,10-15]. In this context, data fusion helps reducing data amount, maintaining data faults, and extracting useful data [16]. Data fusion could be achieved in IoT applications at three scales: (1) low-scale, which uses the acquired sensors data directly in the fusion process; (2) middle-scale, where certain extracted features from the heterogeneous IoT data are used in the fusion process; and (3) high-scale, which decides the most optimum decision from multiple decisions [17]. Accordingly, several combinations of the data fusion levels can be utilized for data prediction. [5]. However, our scope focused on the low data fusion scale which immediately fuses data from IoT data sources. In this context, the maintained IoT fused data can be used for data prediction, avoiding IoT data processing challenges. Hence, this paper proposes the domain-independent Data Fusion for Data Prediction (DFDP) approach that maintains many features of IoT data based on data input-data output fusion approach. DFDP approach aims to improve IoT data prediction accuracy, reliability, and processing time.

This paper contributes the following added values. It:

1. Investigates the challenges of different data prediction applications at various IoT domains.
2. Presents the Data Fusion for Data Prediction (DFDP) approach as a generic IoT-based data prediction approach for any smart city application.
3. Ensures DFDP data prediction processing time robustness by reducing IoT data size.
4. Ensures DFDP data prediction reliability by maintaining IoT data faults.
5. Ensures DFDP data prediction accuracy by considering IoT data spatiotemporality and freshness.
6. Examines DFDP using datasets of different fields of smart cities.

The organization of this paper is: Section 2 overviews multiple approaches used to predict data in different IoT applications. Section 3 presents the suggested DFDP approach, showing the clear clarification of its tiers. Section 4 debates DFDP experiments. Finally, the conclusion section concludes DFDP findings and presents our upcoming work.

## 2. Related Works

Numerous approaches for predicting data have been developed to the IoT-based systems. In this section, we investigated the employed data prediction techniques and emphasized the approaches' gaps. The approach proposed in [18] was a data prediction approach for predicting human activities on a daily basis. Authors utilized users' mobile telecommunications and human body sensors' data to deduce human behavior patterns and predict his activities. The proposed approach was based on DT technique reaching a 75% accuracy level. However, the approach missed considering IoT massive size, IoT data temporality and freshness while analysis. In [19], authors proposed an approach for predicting and recommending video content based on IoT user profiles. The video metadata content was used to create the associated user profile from the continuously tracked user communications with the system. It was based on Artificial Neural Networks (ANN) using item content and user ratings data. Although it reached a 90% accuracy level, it ignored videos meta-data spatiality, freshness, and validity. The approach in [20] was a context aware prediction system for IoT-based social media activities. It infers the contextual preferences by reading context, user profile, and social media activities using NB. Analyzing all the acquired IoT data without data reduction and ignoring data freshness and temporality were the main drawbacks. For IoT-based web mining, the authors suggested a method for recommending web surfers preferred pages in [21]. The clustered Markov model was utilized in the suggested method for making predictions, which aimed to cluster web documents according to web services before predicting web pages using Markov model to reach 90% accuracy level. Yet, ignoring web pages meta-data freshness and surfing data temporality were the main approach's issues. In smart transportation IoT domain, authors in [22] presented an IoT-based approach for accident rates prediction using ANN model. Different numbers were used by the model such as vehicles, accidents, and population. Although the experimental results illustrated that 85% accuracy level was reached, neither considering data freshness nor data reduction were the main missing. Table 1 summarizes the mostly used approaches for predicting data in IoT applications according to the applied IoT domain, used technique, the limitations of the approach, and the applied evaluation metric.

Table 1 Evaluation Summary of the main IoT-based data prediction approaches

| Ref | IoT domain | Technique | Limitations | Evaluation metric |
|---|---|---|---|---|
| [18] | Smart homes | DT | Considering IoT data temporality, massive size, and freshness | 75% Accuracy level |
| [19] | IoT data streaming | ANN | Considering IoT data spatiality, freshness, and validity | 90% Accuracy level |
| [20] | IoT-based social media | NB | Maintaining IoT data massive size, freshness, and temporality | - |
| [21] | IoT-based web mining | Markov model | Maintaining IoT data freshness and temporality | 90% Accuracy level |
| [22] | Smart transportation | ANN | Considering IoT data massive size and freshness | 85% Accuracy level |

## 3. Data Fusion for Data Prediction (DFDP) Approach Demonstration

A complete clarification of the Data Fusion for Data Prediction (DFDP) approach is demonstrated herein. Figure. 1 presents the DFDP architecture, that utilizes two tiers: *Data Manager* and *Data Fusion Manager*. DFDP is designed to process the structured IoT data on any processing model. Another two tiers: (1) *Domain-based analysis* and (2) *Data presentation* are not considered as DFDP modules, because they are dependent on the IoT application and specific to the business use case. A detailed clarification of the main tiers of DFDP are clarified in the next subsections.

### 3.1. Data Manager

This module starts the DFDP processing scenario, which reads the raw structured IoT data to directly handle multiple IoT data characteristics before the data fusion and prediction stages through the following modules.

- **Data Freshness Handler:** Data freshness depends on the IoT application domain. For instance, traffic-related data are considered fresh for minutes, while in smart energy IoT domain, data are considered fresh for months [23]. Hence, this STDF module manages data freshness through testing their generation time $GT$ if exceeds a specific freshness time duration $FT$ dependent on the IoT domain [5]. The data rows $XD$ of data source $j$ with a generation time $GT_j$ exceeds the freshness time duration $FT$ are ignored as shown in Eq. (1):

$$XD_j = GT_J \geq FT \qquad (1)$$

where $XD_j$ are the discarded data rows of data source $j$, $GT_j$ is the data rows generation time at data source $j$, and $FT$ is the specific freshness time duration.

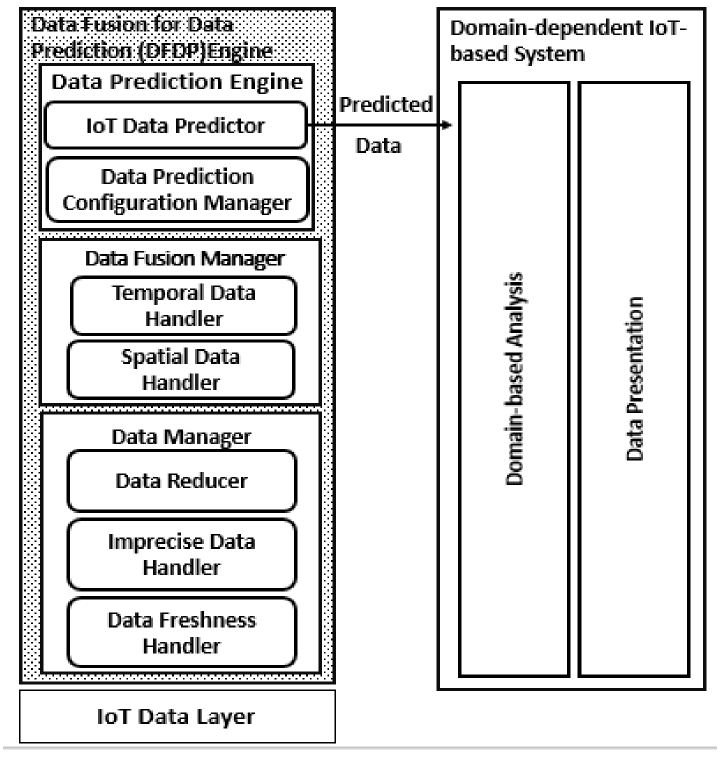



Figure. 1: The system architecture of the proposed Data Fusion for Data Prediction (DFDP) approach on a domain based IoT application

- **Imprecise Data Handler**: IoT data are faulty and noisy because of the unreliable data sources [24]. Thus, prior to data prediction, it is crucial to maintain IoT data missing and faults [25]. Therefore, this sub-module checks each data parameter to replace faulty one with the parameter mean value $Mn_i$. For instance, for $X$ numerical parameter, DFDP generates $(Mn_1, Mn_2, \dots, Mn_i, \dots . Mn_X)$ as shown in Eq. (2) [26]:

$$Mn_i = \frac{1}{n}\sum_{j=1}^{j=n} Z_{ji} \qquad (2)$$

where $Mn_i$ represents the mean of parameter $i$, $n$ represents the count of data rows of parameter $i$, and $Z_{ji}$ represents the parameter's value $i$ at the data row $j$. Additionally, to process non-numeric input for additional mathematical computations at the data prediction step, several encoding techniques are utilized to change the data format into a numeric format. [27].

- **Data Reducer:** The enormous IoT data volume which negatively affects the data prediction results are managed herein [28]. Thus, this STDF sub-module reduces data volume by applying two phases of cluster sampling technique. Where, data are clustered based on the same data source ($SID$) then instead of choosing all elements from one cluster, it select random sample of all clusters [28, 29]. Thus, it horizontally reduces the count of IoT data rows per each source [5].

### 3.2. Data Fusion Manager

Handling the temporal and spatial IoT data characteristics is the responsibility of this tier. It performs the data fusion procedure on the final data rows from the *Data Manager* tier through the following modules.

- **Spatial Data Handler:** IoT data spatiality is preserved through grouping IoT data rows according to their location ID ($LID$) by utilizing K-means algorithm [5]. It receives the data rows after reduction and clustering based on their $SID$, then it groups all data rows of the same $SID$ based on their $LID$ (centroids) by performing one stage of K-means technique and the constant centroids [30, 31, 32].
- **Temporal Data Handler:** IoT data temporality is preserved by aggregating each $SID$ data rows per $LID$ by utilizing Tiny AGgregation (TAG) data aggregation algorithm[5]. Hence, using the resultant spatial corelated grouped data rows, it aggregates all $SID$ data rows to select the most fresh data row per $LID$ which has the minimum $GT$ [33].

### 3.3. Data Prediction Engine

This tier performs the data prediction process after maintaining IoT data freshness, quality, size and spatiotemporality, which ensures the effective data prediction by using fresh, cleansed, reduced and spatially and temporally correlated data. Data prediction is performed using K Nearest Neighbor (KNN) technique through the following sub-modules.

- **Data Prediction Configuration Manager:** This sub-module prepares IoT data to enable their processing using KNN by utilizing encoding methods to convert the data format from non-numeric to a numerical [34]. For example in nominal parameters, label encoding is used and each label is mapped to an integer value [35]. Also, it standardizes parameter values in case of parameter values are measured in different units [36]. For example, if height parameter is measured in centimeters and meters, it standardizes all values to the same unit. Furthermore, basic properties are determined in this module to enable the KNN, such as the $K$ value [37].

- **IoT Data Predictor:** This sub-module predicts the unknown parameter's value of specific data row using KNN technique which use the same parameter's values of the nearest neighbor data rows [38]. DFDP constructs a minimal dataset for each $SID$ using the resultant location-based aggregated data rows with $M$ parameters. Then, it determines the distance between the current data row $R$ and each neighbor data row $NR_i$ using the Euclidean distance $(Ed)$ equation as shown in Eq.(3) [39]. Neighbors data rows are ascendingly sorted based on their distance, and the $K$ neighbor data rows with minimum distances are selected to predict the unknown parameter value $PR_j$ via the average equation as shown in Eq.(4) [40] in case of continuous parameter values.

$$\boldsymbol{Ed\ (R, NR_i)} = \sqrt{\sum_{\boldsymbol{j=1}}^{\boldsymbol{j=M}} \left(\boldsymbol{R_j - NR_{ij}}\right)^{\boldsymbol{2}}} \qquad (\boldsymbol{3})$$

where $\boldsymbol{R_j}$ is the parameter $\boldsymbol{j}$ value in current data row $\boldsymbol{R}$, $\boldsymbol{NR_{ij}}$ is the parameter $\boldsymbol{j}$ value in neighbor data row $\boldsymbol{i}$, and $\boldsymbol{M}$ represents the parameters count in all data rows.

$$\boldsymbol{PR_j} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{i=K}} \boldsymbol{NR_{ij}} \Big/ \boldsymbol{K} \qquad (\boldsymbol{4})$$

where $PR_j$ is the parameter $j$ predicted value in current data row $R$, $NR_{ij}$ is the parameter $j$ value in neighbor data row $i$, and $K$ represents the neighbors data rows count. Otherwise, for the nominal parameter values, DFDP uses majority voting instead of averaging that computes proportion $P$ for each value in the $K$ data rows as demostrated in Eq.(5) [11]:

$$P = B/K \qquad (5)$$

where $B$ represents the value's occurrences count in the $K$ data rows and $K$ represents the neighbor data rows count. DFDP considers the predicted value is the one which has the maximum $P$ value.

## 4. Experimentation

This section demonstrates the experiments criteria, evaluation metrics and the used datasets while evaluating DFDP modules. The next subsections demonstrate a thorough insight of the experimentation process and the utilized datasets.

### 4.1. Experimental process

The experiments performed to ensure DFDP efficiency are demonstrated herein. Using 2.70GHz core i7, 8GB RAM, and 1T hard disk for hardware specifications to carry out the experiments for DFDP. The experiments were designed to assess how IoT data prediction reliability, robustness, and accuracy are impacted by IoT data characteristics. DFDP processing begins with checking the freshness, pre-processing, and reducing the collected IoT data rows in the *Data Manager* using two different freshness time durations: 10 seconds 24 hours for S1 and S2 respectively and 200 and 500 sample sizes for dataset1 and dataset2. IoT data are then grouped by the *Spatial Data Handler* based on location IDs and the *Temporal Data Handler* aggregates the data rows in each group to the freshest data row. Then, the *IoT Data Predictor predicts* the unknown data using the aggregated IoT data rows. DFDP performance is evaluated via tracking IoT data freshness, reliability, size, spatiality, temporality, processing time and data prediction accuracy. Many evaluation standards are considered to evaluate DFDP such as the standard deviation ($SD$) for IoT data reliability as shown in Eq. (6) [41] and the Root Mean Square Error ($RMSE$) for IoT data prediction accuracy as shown in Eq. (7) [42]:

$$\mathrm{SD_i} = \sqrt{\sum(\mathrm{X_i} - \mathrm{Mn_i})^2/M} \qquad (6)$$

where $SD_i$ is the parameter $i$ standard deviation, $Mn_i$ is the parameter $i$ mean value, $X_i$ is the parameter $i$ data row value and $M$ represents the data rows count in the dataset.

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{N}(x_i - y_i)^2} \qquad (7)$$

where $M$ represents the data rows count in the dataset, $x_i$ represents the original data, and $y_i$ represents the corresponding predicted data to $x_i$.

## 4.2. Datasets

Two datasets of various IoT applications are utilized to assess the DFDP performance. DFDP minimizes the amount of IoT data regardless of dataset volume. For instance, an IoT traffic dataset (S1), represents data rows related to the smart transportation field [43]. S1 is 843KB, represented by seven parameters: ID, location ID, owner, state, type, date-time, entry-dist. All nominal parameters, such as: owner, state and type are encoded in the *Data Prediction Configuration Manager*. Second, an IoT water consuming dataset (S2), represents data rows related to smart urban planning field [43]. S2's size is 960 KB, represented by 11 parameters: hydrant-number-model, owner, ID, node-number, type, dist_to_valve, branch_valve_model, state, project-number, location ID, and date-time. All nominal parameters like owner, state and type are encoded in the *Data Prediction Configuration Manager*.

## 5. Discussion

This section presents our experiments that assessed DFDP effectiveness to predict IoT data as follows: (1) Managing data freshness using different user-defined freshness time durations, (2) Ensuring data quality by managing IoT data errors on the different datasets, (3) Reducing the huge data on different datasets, (4) Preserving data spatiality by monitoring data rows location IDs on different datasets, (5) Managing data temporality by monitoring the aggregated data rows at different datasets, (6) Monitoring the consumed processing time before and after reducing data to ensure the robustness of data prediction, and (7) Ensuring data prediction accuracy at the different datasets. The carried-out experiments with their results are discussed in the next subsections.

## 5.1. Data Freshness Evaluation

Managing IoT data freshness is evaluated in this experiment by checking the fresh data rows count having an accepted $GT$ on different time duration. Thus, we investigated the fresh data rows count of S2 at 1, 2 and 3 days durations. As presented in Figure. 2, 10,020, 16,320 and 20,170 fresh data rows were passed at the three durations respectively, which demonstrates that the fresh data rows count increases by increasing the freshness time duration.

## 5.2. Handling IoT Data Quality

Handling IoT data reliability is evaluated in this experiment by checking the $SD$ for the generated mean value in S1 and S2. Figure. 3 shows the $SD$ of entry-dist and dist_to_valve (6.8, 4.3) for S1 and S2 respectively, reaching maximum error level at 6.8, proving 93.2% quality level for the used data rows.

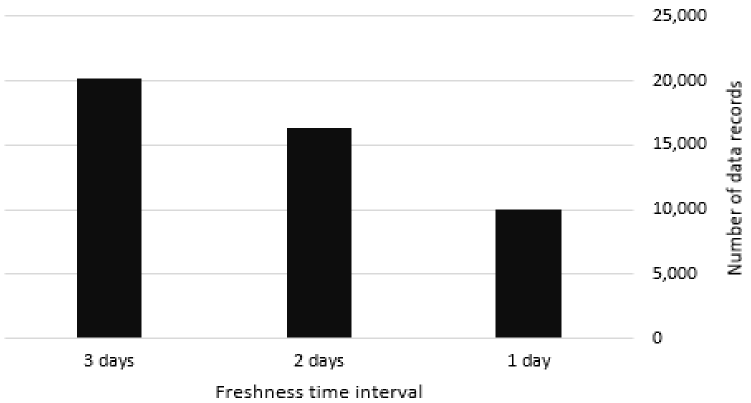



Figure. 2 Evaluating IoT data freshness at different freshness time durations

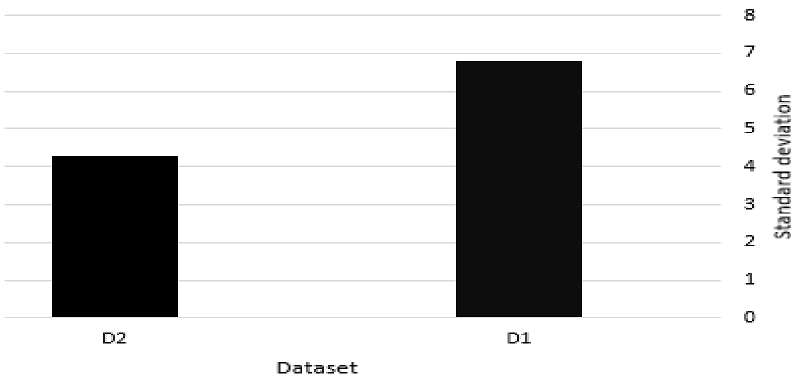



Figure. 3 SD values of parameters means at S1 and S2

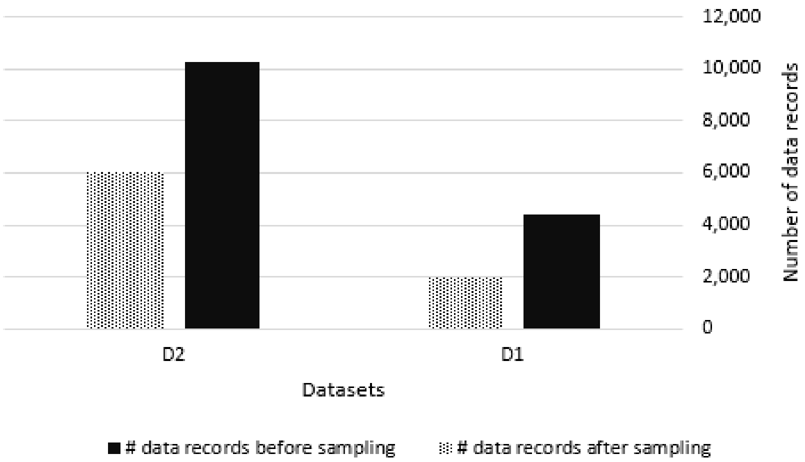



Figure. 4 Evaluating IoT data reduction at S1 and S2

### 5.3. Handling IoT Data Size

IoT data reduction is evaluated in this experiment by checking the sampled data rows count at S1 and S2. Thus, the sampled data rows count is tested using the sample size of data rows: 200 and 500 for S1 and S2 respectively. Figure. 4 shows the data rows count in both datasets before and after sampling. There were 4,400 and 10,300 data rows per day for S1 and S2 before sampling, while 2,000 and 6,000 data rows per day were found after sampling for S1 and S2. This proves that DFDP efficiently reaches an average 50% of data reduction using 50%–60% of the population as sample size.

### 5.4. Handling IoT Data Spatiality

Preserving IoT data spatiality is evaluated in this experiment through checking location IDs number in different datasets. Thus, the number of LID per each owner ***(***SID***)*** is tracked in each dataset. As shown in Figure. 5, 10 and 12 $SID$ are found for S1 and S2 respectively, and average 3 and 6 $LID$ are found per each $SID$ for S1 and S2. This experiment proves that the larger the size of dataset, the more location IDs used to cluster the data rows.

### 5.5. Handling IoT Data Temporality

Preserving the IoT data temporality is evaluated in this experiment by checking $LID$ and aggregated data rows numbers on a time-based specific $SID$ to examine the aggregation effectiveness. Each $SID$ has an average 200 data rows (after sampling) are generated from different number of $LID$. Thus, we investigated the results for $SID_1$ at S1 over three seconds of simulation. Table 2 shows that the data rows are disseminated from 3 & 2, and 3 $LIDs$ at the first, second, and third seconds of execution. Hence, the gathered data rows of $SID_1$ in S1 are 3, 3 and 2 data rows respectively.

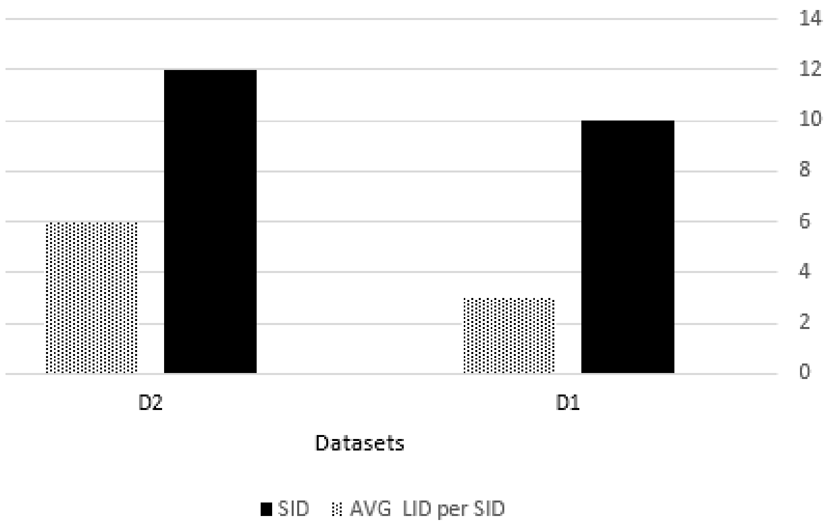



Figure. 5 The number of $\boldsymbol{SID}$ and $\boldsymbol{LID}$ at S1 and S2

### 5.6. The Continuous IoT Data Processing Evaluation

This test tracks the consumed processing time of the Data Prediction Engine before and after utilizing both sampling and aggregation data reduction techniques in Data Manager and Data Fusion Manager using both datasets preserving the continuous processing of IoT data. As shown in Figure. 6, 34 and 12 seconds were consumed for S1 before and after using the reduction techniques. In addition, 56 and 25 seconds were consumed for S2 before and after using the reduction techniques. This demonstrates that DFDP preserves the consumed processing time by an average of 53%.

Table 2 Three seconds of simulation of $\boldsymbol{SID_1}$ in S1 at the *Temporal Data Handler*

| Seconds | Number of $LID$ | Number of data rows per $LID$ $LID_1$ | $LID_2$ | $LID_3$ | Number of aggregated data rows per $SID_1$ |
|---|---|---|---|---|---|
| 1 | 2 | 98 | 72 | 30 | 3 |
| 2 | 3 | 78 | 65 | 57 | 3 |
| 3 | 3 | - | 134 | 66 | 2 |

## 5.7. Evaluating IoT Data Prediction Accuracy

The accuracy of DFDP IoT data prediction is evaluated in this experiment after maintaining IoT data features using DFDP modules. We randomly selected the predicted data rows to check the $\boldsymbol{RMSE}$ of in ($\boldsymbol{LID_1}$, $\boldsymbol{LID_2}$) of $\boldsymbol{SID_1}$ at S1 and S2 using $\boldsymbol{K}$=3. In S1, the $\boldsymbol{RMSE}$ of the predicted data rows for the attributes: type, status, owner, and entry-dist, is computed based on the entry-dist attribute only, since the rest of the attributes are nominal with P instead of the mean. In S2, the $\boldsymbol{RMSE}$ of the predicted data rows for the attributes: node-number, type, hydrant-number-model, dist_to_valve, branch_valve_model, owner, state, project-number, is computed based on the dist_to_valve attribute only, since the rest of the attributes are nominal with P instead of the mean. In S1, the $\boldsymbol{RMSE}$ of the predicted data row at both location IDs were: 8.2 and 7.6 for $\boldsymbol{LID_1}$ and $\boldsymbol{LID_2}$ respectively. As for S2, the $\boldsymbol{RMSE}$ of the predicted data row at both location IDs were: 5.4 and 6.2 for $\boldsymbol{LID_1}$ and $\boldsymbol{LID_2}$ respectively, which implies the accurate prediction results for larger datasets. Figure. 7 shows that DFDP reaches the maximum $\boldsymbol{RMSE}$ value at 8.2, which ensures the accurate IoT data prediction with 91.8% accuracy level.

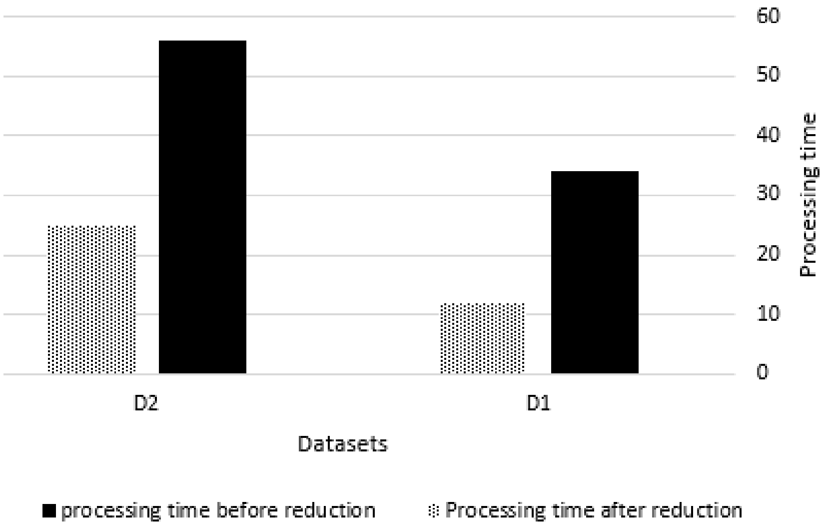



Figure. 6 The consumed processing time before and after DFDP data reduction at S1 and S2

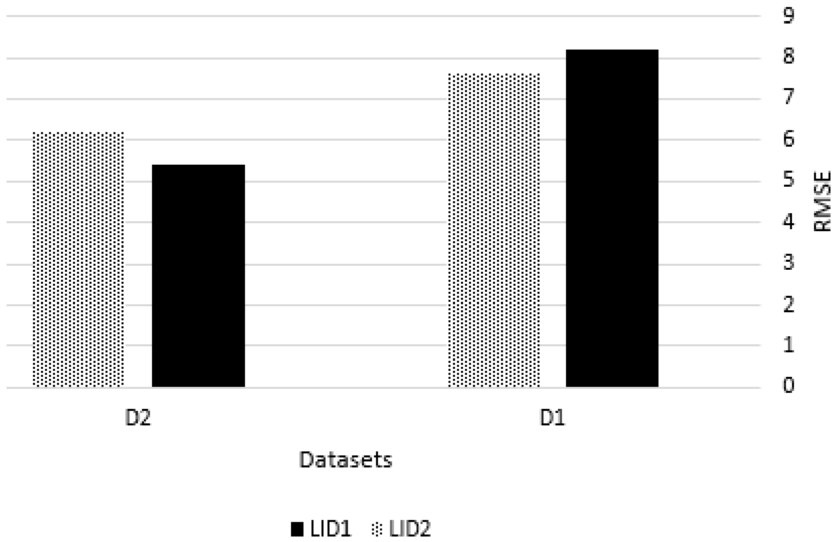



Figure. 7 DFDP data prediction accuracy at S1 and S2

## 6. Conclusion

Recently, data prediction had a significant demand in IoT applications. It faces multiple challenges due to IoT data characteristics like size, quality, spatiotemporality, freshness etc. In this study, Data Fusion for Data Prediction (DFDP) approach is uniquely prompted as a data prediction approach that maintains different IoT data features by utilizing data fusion on three processing stages: (1) Data manager, (2) Data fusion manager, and (3) Data prediction engine, irrespective of the IoT domain and coping any business functions. Firstly, it processes the acquired IoT data to manage most of the introduced IoT data features, like, data freshness, quality, and size. Secondly, it maintains the spatiotemporal IoT data characteristics using data aggregation as a main data fusion technique. Finally, the third stage is responsible for data prediction, where the KNN is used as the main data prediction technique. The experimental results indicate the optimum performance of DFDP in different aspects, as: (1) ensuring the totally fresh IoT data, (2) managing IoT data impreciseness by 93.2% accuracy level, (3) achieving an average 50% and 53% of data reduction and processing time reduction respectively (4) accurately predicts IoT data with 91.8% accuracy rate. Our future work is focusing on examining the DFDP approach for multiple types of software testing in large and distributed frameworks, such as the multi-agent environments [44, 45], IoT-based systems [46-53] and service-oriented systems [54, 55]. We also plan to improve the DFDP approach via considering IoT data heterogeneity by supporting different data structures.

## References

1. R. Mazhar, et al., Urban planning and building smart cities based on the internet of things using big data analytics, Comput. Networks. 101 (2016) 63–80. https://doi.org/10.1016/j.comnet.2015.12.023.
2. F. Alam, et al., Data fusion and IoT for smart ubiquitous environments: A survey, IEEE Access. 5 (2017) 9533–9554. https://doi.org/10.1109/ACCESS.2017.2697839.
3. F. Azzedin, et al., Internet-of-Things and information fusion: Trust perspective survey, Sensors. 19 (2019) 1929. https://doi.org/10.3390/s19081929.

4. A. Taivalsaari, et al., A Roadmap to the Programmable World: Software Challenges in the IoT Era, IEEE Softw. 34 (2017) 72–80. https://doi.org/10.1109/MS.2017.26.
5. D. Fawzy, et al., The Spatiotemporal Data Fusion (STDF) Approach: IoT-Based Data Fusion Using Big Data Analytics, Sensors. 21 (2021) 7035.
6. D. Fawzy, et al., The Internet of Things and Architectures of Big Data Analytics: Challenges of Intersection at Different Domains, IEEE Access. (2022).
7. M. Meraj, et al., A critical review of detection and prediction of infectious disease using IOT sensors, in: Second Int. Conf. Electron. Sustain. Commun. Syst., 2021: pp. 679–684.
8. D. Fawzy, et al., The Evolution of Data Mining Techniques to Big Data Analytics: An Extensive Study with Application to Renewable Energy Data Analytics , ISSN: 2321–0893, vol. 4, no. 3, pp, Asian J. Appl. Sci. 4 (2016) 756–766.
9. A. Zanella, et al., Internet of things for smart cities, IEEE Internet Things J. 1 (2014) 22–32. https://doi.org/10.1109/JIOT.2014.2306328.
10. X. Qin, et al., Data fusion in the Internet of Things, Procedia Eng. 15 (2011) 3023–3026. https://doi.org/10.1016/j.proeng.2011.08.567.
11. D. Fawzy, et al., The Spatiotemporal Data Reduction (STDR): An Adaptive IoT-based Data Reduction Approach, in: 2021 Tenth Int. Conf. Intell. Comput. Inf. Syst., 2021: pp. 355–360. doi: 10.1109/ICICIS52592.2021.9694199.
12. D. Fawzy, et al., “Trio-V Wind Analyzer: A Generic Integral System for Wind Farm Suitability Design and Power Prediction Using Big Data Analytics”, Journal of Energy Resources Technology, JERT, 140(5), 051202, 2017, doi: 10.1115/1.4038119.
13. D. Fawzy, et al., An IoT-based resource utilization framework using data fusion for smart environments. Internet of Things, 21, p.100645, doi: 10.1016/j.iot.2022.100645.
14. D. Fawzy, et al., “An Approach for Erosion and Power Loss Prediction of Wind Turbines Using Big Data Analytics”, In: Woon W., Aung Z., Kramer O., Madnick S. (eds) Data Analytics for Renewable Energy Integration: Informing the Generation and Distribution of Renewable Energy. DARE 2017. Lecture Notes in Computer Science (LNCS), Vol 10691, pp. 33-46, Springer, Cham, 2017, doi: 10.1007/978-3-319-71643-5_4.
15. D. Fawzy, et al., “WFEC: Wind Farms Economic Classifier Using Big Data Analytics”, In the Proceedings of the Eighth IEEE International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 154-159, 2017, doi: 10.1109/INTELCIS.2017.8260046.
16. T. Meng, et al., A survey on machine learning for data fusion, Inf. Fusion. 57 (2020) 115–129. https://doi.org/10.1016/j.inffus.2019.12.001.
17. P. Rashinkar, et al., An overview of data fusion techniques, in: Int. Conf. Innov. Mech. Ind. Appl., 2017: pp. 694–697.
18. A. Papliatseyeu, et al., Mobile habits: Inferring and predicting user activities with a location-aware smartphone, in: 3rd Symp. Ubiquitous Comput. Ambient Intell., 2009: pp. 343–352.
19. D. Silva, et al., Personal TVware: an infrastructure to support the context-aware recommendation for personalized digital TV, Int. J. Comput. Theory Eng. 4 (2012) 131.
20. M. Fiedler, et al., Influence of community design on user behaviors in online communities, J. Bus. Res. 67 (2014) 2258–2268.
21. N. Sharma, Strategies for Mining Web Logs: A Survey, Int. J. Eng. Res. Appl. 4 (2018) 21–27.
22. M.D. Angeline, Association rule generation for student performance analysis using apriori algorithm, SIJ Trans. Comput. Sci. Eng. 1 (2017) 12–16.
23. M.A. Jan, , et al., An AI-enabled lightweight data fusion and load optimization approach for Internet of Things, Futur. Gener. Comput. Syst. 122 (2021) 40–51.

24. S. Pasricha, et al., A survey on energy management for mobile and IoT devices, IEEE Des. Test. 37 (2020) 7–24.
25. Z. Yan, et al., Fusing and mining opinions for reputation generation, Inf. Fusion. 36 (2017) 172–184. https://doi.org/10.1016/j.inffus.2016.11.011.
26. Q. Zhu, et al., Thompson sampling algorithms for mean-variance bandits, in: Int. Conf. Mach. Learn., 2020: pp. 11599–11608.
27. F. Ridzuan, et al., A review on data cleansing methods for big data, Procedia Comput. Sci. 161 (2019) 731–738.
28. Deep learning for IoT big data and streaming analytics: A survey, IEEE Commun. Surv. Tutorials. (n.d.).
29. S. Wang, et al., K-means clustering with incomplete data, IEEE Access. 7 (2019) 69162–69171.
30. M.Z. Hossain, et al., A dynamic K-means clustering for data mining, Indones. J. Electr. Eng. Comput. Sci. 13 (2019) 521–526.
31. H. Zhang, et al.  PSO and K-means-based semantic segmentation toward agricultural products, Futur. Gener. Comput. Syst. 126 (2022) 82–87.
32. M.K. Gupta, et al., An empirical evaluation of K-means clustering algorithm using different distance similarity metrics, in: Proc. ICETIT 2019, Springer, Cham., 2020: pp. 884–892.
33. B. Pourghebleh, et al., Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research, J. Netw. Comput. Appl. 97 (2017) 23–34.
34. D. Auge, et al., A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks, Neural Process. Lett. (2021) 1–18.
35. A. Jain, et al., Gist: Efficient data encoding for deep neural network training, in: 2018 ACM/IEEE 45th Annu. Int. Symp. Comput. Archit., 2018: pp. 776–789.
36. S. Mittal, et al., A survey of encoding techniques for reducing data-movement energy, J. Syst. Archit. 97 (2019) 373–396.
37. M. Strohbach, et al., Towards a Big Data Analytics Framework for IoT and Smart City Applications, Model. Optim. Sci. Technol. 4 (2015) 257–282. https://doi.org/10.1007/978-3-319-09177-8-11.
38. R. Zhu, et al., KNN-based approximate outlier detection algorithm over IoT streaming data, IEEE Access. 8 (2020) 42749–42759.
39. M.A. Khan, et al., A deep learning-based intrusion detection system for mqtt enabled iot, Sensors. 21 (2021) 7016.
40. L.G. Maxim, et al., Euclidean distance degree of the multiview variety, J. Appl. Algebr. Geom. 4 (2020) 28–48.
41. J. Ching, et al., Constructing quasi-site-specific multivariate probability distribution using hierarchical Bayesian model, J. Eng. Mech. 147 (2021) 04021069.
42. M. Ahmad, et al., Development of prediction models for shear strength of rockfill material using machine learning techniques, Appl. Sci. 11 (2021) 6167.
43. Available online: https://data.world/datasets/smartcity (accessed on February- 2022), (2022).
44. S. Moussa, et al., “Integrating Encrypted Mobile Agents with Smart Spaces in a Multi-agent Simulator for Resource Management”, Journal of Software (JSW), Special Issue in Information Education and Technological Application, 5(6), pp.630-636, Academy Publisher, June 2010, doi: 10.4304/jsw.5.6.630-636.
45. S. Moussa, et al., “A Multi-agent Simulator for Resource Management in Smart Spaces”, IEEE 2009 Joint Conferences on Pervasive Computing (JCPC), pp. 765 - 770, Taiwan, December 2009, doi: 10.1109/JCPC.2009.5420081.

46. P. Kandil, et al., "A Study for Regression Testing Techniques and Tools", International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 5, No. 4, pp. 64-84, 2015, doi: 10.7321/jscse.v5.n4.1.
47. P. Kandil, et al., “A Methodology for Regression Testing Reduction and Prioritization of Agile Releases”, IEEE 2015 International Conference on Information and Communication technology and Accessibility (ICTA 2015), pp. 1-6, Marrakesh, Morocco, December 2015, doi: 10.1109/ICTA.2015.7426903.
48. R. Elghondakly, et al., “A Comprehensive Study for Software Testing and Test Cases Generation Paradigms”, ACM Proceedings of International Conference on Internet of things and Cloud Computing (ICC 2016) University of Cambridge, UK, March 2016, doi: 10.1145/2896387.2896435.
49. R. Elghondakly, et al., “An Optimized Approach for Automated Test Case Generation and Validation for UML diagrams”, Asian Journal of Information Technology (AJIT), 15(21), pp. 4276-4290, Medwell Journals, 2016, doi: 10.3923/ajit.2016.4276.4290.
50. P. Kandil, et al., "Regression Testing Approach for Large-Scale Systems", IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 132-133, 2014, doi: 10.1109/ISSREW.2014.96.
51. N. Medhat, et al., "A Framework for Continuous Regression and Integration Testing in IoT Systems based on Deep Learning and Searchbased Techniques." IEEE Access, Vol. 8, pp. 215716-215726, 2020. doi: 10.1109/ACCESS.2020.3039931.
52. N. Medhat, et al., "Enhancing Test Cases Prioritization for Internet of Things based systems using Search-based Technique." International Journal of Intelligent Computing and Information Sciences 21, no. 1 (2021): 84-94. doi: 10.21608/ijicis.2021.69462.1076.
53. N. Medhat, et al., "Testing Techniques in IoT-based Systems", 2019 IEEE Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, pp. 394-401, IEEE, 2019. doi: 10.1109/ICICIS46948.2019.9014711.
54. R. Elghondakly, et al., “Handling Faults in Service Oriented Computing: A Comprehensive Study”, The 2020 International Conference on Computational Science and Its Applications (ICCSA 2020), Cagliari, Italy, Springer LNCS, 2020. doi: 10.1007/978-3-030-58811-3_67.
55. R. A. ElGhondakly, et al., "Mutual Information-based Modeling for Services Dependency." IEEE Transactions on Services Computing, pp. 1 - 18, 2022. doi: 10.1109/TSC.2022.3207232.