Context Aware Data Fusion on Massive IOT Data in Dynamic IOT Analytics

S.S. Saranya
Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Kattankulathur, Chennai. E-mail: saranyas6@srmist.edu.in

Dr.N. Sabiyath Fatima
Associate Professor, Department of Computer Science and Engineering, B.S.Abdur Rahman Crescent Institute of Science and Technology, Vandallur, Chennai.
E-mail: sabiyathfatima@crescent.education

Received November 04, 2020; Accepted December 10, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V17I2/WEB17080

Abstract

Educational Data management is a critical task for the researchers due to mammoth data generated by sensors and IoT (Internet of Things) devices. Managing this huge volume of data, cleaning this data from impurities is an inherent need. DF (Data Fusion) processes combine data from multiple sources based on their similarity for an easy management. DF processes focus on many factors like nature of data and application that uses that data. Many DFAs (Data Fusion approaches) have been proposed without detailing on the context for integrating data in fusion tasks. This work attempts to cover this gap of context's relevance by proposing a technique CDFT (Context aware Data Fusion technique). In this research work, initially data from IoT devices will be gathered and pre-processed to make it clear for the fusion processing. In this work, boundary based noise reduction algorithm is introduced for data pre-processing which attempts to label the unlabelled attributes in the data's that are gathered, so that data fusion can be done accurately. After pre-processing Context aware data fusion is performed which will combine the data's from multiple IoT devices together with the concern of context. Finally this combined data will be learnt using the convolution neural network for data fusion performance checking. The proposed CDFT is simulated on Matlab whose results prove that the proposed technique obtains optimal outcomes.

Keywords

Data Fusion, Context Attributes, IoT Devices, Heterogeneous Data, Noise Reduction, Unlabelled Data.

Introduction

IoT paradigm has gathered significant attention in industries and research for a decade [1] mainly due to its capabilities. IoTs can a new world from Internet connected objects that communicate automatically [2]. These devices facilitate the human vision of connectivity anywhere to anyone or from any place [3]. The European Union defines this same vision as “The IoT allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any network and Any service” [4].

IoT operations are mainly dependent on Sensor networks [5] where sensors are small devices that record any changes occurring in its surroundings like humidity or temperature changes. Physically a sensor node can congregate many sensors and communicate and process sensed data [6]. Networks based on sensors communicate between them using a wireless or wired connection [7]. They can be classified as heterogeneous or homogeneous. Internet is the backbone for connecting these sensor networks using varied techniques.

Sensor nodes are in small sizes and cost very less and are deployed randomly in a dense manner in places like forests where monitoring changes are required [8]. Sensor networks have been researched for long even before the arrival of IoT [9] as seen from literature. After the emergence of IoT, sensors have been adopted as the technology to apprehend IoT. Current rise in the capabilities of SMDs (Smart Mobile Devices) like Mobiles, Tablets and watches has made them substitute sensors.

Revolutionary applications like Smart city are possible with these sensors and SMDs [10]. IoT environments can have multiple sources of data where a fusion of these data for developing innovative services becomes a complex issue [11]. Smart city applications involve billions of devices which carry sensed information. This information has to be fused automatically which is a major challenge in such applications. Single sensors carry limited information but when many sensors data is analyzed it can portray the required information about environmental changes [12]. This data consolidation from multiple sensors increases accuracy in the sensed data [13]. For example, the use of RFID tags in supermarkets can trace thefts. When two sensors send their sensed data about the same location, their average reading is more accurate. This also reduces the volume of data to be processed as two different streams are combined.

The main goal of this research work is to manage the data’s that are gathered from the multiple IoT devices that are integrated in the educational field. This research work attempt to group the dataset that are gathered from the educational platform to assess the activity of the students. This is attained by fuse the collected information to provide the sagetu

Moreover, such cooperated operations are also bases for acquiring new knowledge. Hence, the aim of this research work is to blend data from IoT devices for creating a rich source of information and acquiring knowledge from this combined data. This work also caters to contexts in this data before combining sensed data. Data from multiple IoTs are pre-processed into a data set before being processed. This research work is organized as follows. Section 1 introduces sensors and the need for data fusion. Section 2 is a review of studies related to the research work. The proposed methodology is detailed in section 3 while section 4 analyzes the proposed work with detailed results. The research concludes with section 5.

Related Works

Algorithms for DF were studied by King et al [14]. The study interpreted healthcare application's sensed data in a monitoring context. Their study described in detail about applications in healthcare contexts like Bio-Metric/Health monitoring and stage analysis. The study also details on commercial sensors, their capabilities with comments on gaps that need to be covered or researched.

Qi et al [15] in their work identified trends and issues in IoT DF techniques with discussions about techniques that can overcome their stated issues. PARM (Physical Activity Recognition and Measure) is a key paradigm in many SH (Smart Healthcare) applications where PARM methods rely on design and utilization of ML (Machine Learning) or DF techniques for processing sensed data by eliminating uncertainties and categorizing physical activities.

Time-dependent coefficients in a POD (Proper Orthogonal Decomposition) were compounded by Berry et al [16] in their study. A jet with rectangular multi-stream single expansion ramp nozzle whose core stream operated was used. An underlying deck matched numbers Mi, 1=1..6 with bypass streams as Mi,3=1..0. Schlieren and particle image velocimetry data underwent orthogonal decomposition to get spatial eigen functions. The obtained functions were transformed into their equivalent time-dependent fields of LES (Large-Eddy simulation) for reconstructing POD coefficients temporally.

Application of novel DF technique was evident in the study of Pfeffer et al [17]. They used jSVD (joint Singular Value Decomposition), jCMF (joint Constrained Matrix

Factorization) and SNF (Similarity Network Fusion) for integrating gene expressions and copies numeric data that was applied on the Cancer Genome Atlas UM dataset.

Aerospace platforms autonomy was advanced in the study of Liu et al [18]. They reviewed their twin framework by coupling it with industrial IoT technology. In the proposed study DF techniques played a significant role. Their information flow to high precisions from raw data was done using, sensor-model-model fusion information flow.

A hybrid technique implementing a bi-step fusion process was proposed in the study Traini et al [19].Their technique merged estimations from pattern matching algorithms and applied it on short/long SMD wireless sources before compounding PDR (Pedestrian Dead Reckoning) and RF (Radio Fingerprinting) estimations for overcoming limitations of the used approaches.

Several DF techniques were fused by Martín-Morató et al [20] in their study with an aim to improve AED system's recognition accuracy. They took advantage of the variety in microphone's adverse acoustic conditions. Their evaluation results showed that by using suitable processing schemes AED recognitions could be autonomously increased irrespective of the event location.

An energy aware protocol was proposed by Singh et al [21] in their study. Their technique AIEARP (Artificial Intelligent Energy Aware Routing Protocol) for optimized WSN energy consumptions integrated ANN (Artificial Neural Networks) and KSOM (Kohonen Self-Organizing Map) methods. Clusters were created and re-located in each iteration for effectively distributing node energy and reducing their energy depletions.

Multi-sensor DF technique was discussed by Thupakula et al [22] in their study. Their model identified object in airports for traffic monitoring systems where the object's type, shape and position were identified. Their further research stated collision predictions by retrieving identified object's data and using them as inputs for airport traffic monitoring system.

Context Aware Data Fusion

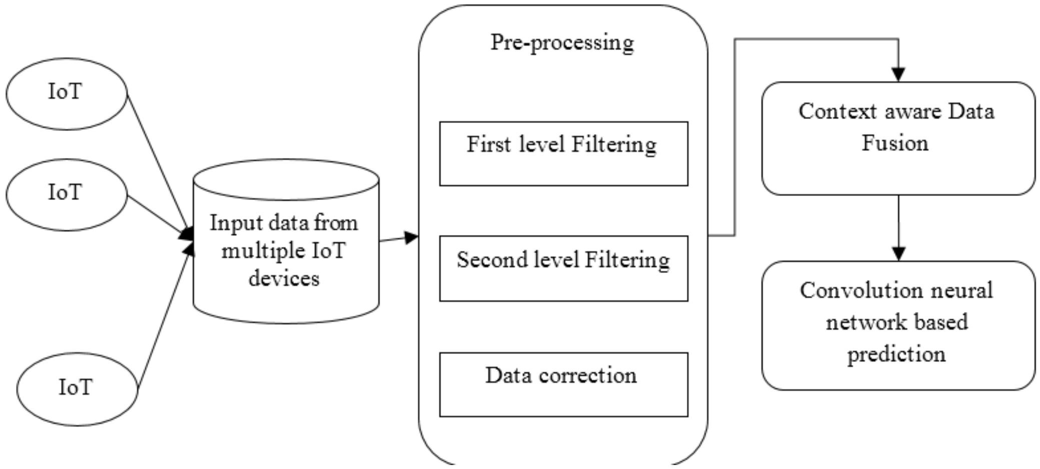

Fig. 1 Process flow of the proposed CDFT

In this research work, data from IoT devices are gathered and pre-processed for its fusion processing. A boundary based noise reduction algorithm is used in data pre-processing which attempts to label unlabelled data attributes for accuracy in DF operations. On pre-processing Context aware DF is performed for combine data from multiple IoT devices. This compounded data is then learnt using CNN (Convolution Neural Network) for checking DF performances. The architecture of the proposed CDFT is depicted in Figure 1 and explained in further subsections.

CDFT Data Pre-processing

Noises or impurities in data are cleared by filtering and correcting them. Filtering identifies wrongly labelled data items which are then corrected using classification models where the models operate on clean data/data sets and may be single classifiers or ensemble models. Cleaned data sets imply data items that remain after cleaning or filtering data/ data sets. Self-Training Corrections (STC) of data occur by operating filter on data sets. These filters are generally classifiers like C4.5 which are operated on the noisy data containing wrongly labelled data. ANC (Adaptive Noise Correction) when applied on data sets creates multiple high quality data set models to clear noises. These data sets are obtained by performing K-fold cross-validations repeatedly. In each iterated round, predicted labels matching corresponding inferred labels results in high quality data sets thus removing noisy parts of the data set. Thus, this work performs preliminary filtering before filters remove noises in data instances. The filters are then trained as a second filtering step. These dual steps of filtering help achieve a noiseless properly labelled clean data for further processing.

Assuming each data instance $x_j$is a multi-label set $l_j=\{l_{jk}\}_{k=1}^{R}$. For binary classifications, Assume $C(+)$ denotes positive labels of the set l_i and $C(-)$ is the no of negative labels in l_i. If pb(+) and pb(-) are positive and negative label probabilities in the set l_i then,

$$\mathrm{pr}(+) = \frac{C(+) + 1}{C(+) + C(-) + 2}$$
$$\mathrm{pr}(-) = \frac{C(-) + 1}{C(+) + C(-) + 2}$$

for a Laplace correction that is applied.

If C(+) is very near to C(-), then the margin between classes |pr(+) - pr(-)| is small. This occurs in two situations namely when items are labelled without proper knowledge. The second case is while labelling complex instances. Hence, for an instance x_i, if |pr(+) - pr(-)| is small, then inference algorithms may not integrate the instance and needs to be filtered. The proposed work uses Algorithm 1 which is listed below. Lines 1 to 7 execute preliminary filtering using |pr(+) - pr(-)|. The second level of filtering is in line no 8, while lines 9 to 13 correct noisy labels and lines 14 to 16 return noiseless data.

Algorithm 1 – Pre-processing Noisy Dataset for Noise Reductions

Input: $\widehat{D} = \{(x_i, \hat{y}_i)\}_{i=1}^{N}$ - a training data set with integrated labels; $\{l_i\}_{i=1}^{N}$ -the multiple label sets of $\widehat{D}$; δ- a threshold

Output: $\widehat{D}$- the corrected data set

1: A-an empty set

2: for i=1 to N do

3: Account the numbers of positive label and negative label in l_i, i.e., N(+) and N(-) respectively

4: Calculate p(+) and p(-)

5:If |p(+)-p(-)|≤ δ

6: The instance i is added to the set A

7: End for

8: A filter is applied to the set $\widehat{D}\backslash A$, and all instances filtered out by the filter comprise a set B

9: $\widehat{D}_c = \widehat{D} \ \backslash(A + B)$

10: Build a classification model f on the set $\widehat{D}_c$

11: for i=1 to sizeof (A+B) do

12: Use the classifier f to relabel the instance i the set A+B

13: End for

14: Update the set A+B to $\overline{\mathrm{A}} + \overline{\mathrm{B}}$ with corrected labels
15: $\widehat{\mathrm{D}} = \widehat{\mathrm{D}}_{\mathrm{c}} + \overline{\mathrm{A}} + \overline{\mathrm{B}}$
16: Return $\widehat{\mathrm{D}}$ as the corrected data set

Context Aware DF

Though the most common approaches used in DF are Kalman filters and HMM (Hidden Markov Models) this work uses DBN (Dynamic Bayesian Network) as it can capture dynamicity in instances by taking into account its past instances for current observations. Further, DBN is a good trade-off for tractability becoming a tool for DF operations. DBNs in this work find influences of context variables on environments without being restricted by probability distributions. They divide data into time slices for representing states of an instance where HMMs are used to discover its observable symptoms. DBN is used mainly to infer states of a given feature of interest and represented by the hidden variable V_t. Updates are performed based on sensory readings and their contexts. $\mathrm{S}_{\mathrm{t}} = (\mathrm{S}_{\mathrm{t}}^{1}, \ldots \mathrm{S}_{\mathrm{t}}^{\mathrm{n}})$ is the set of sensor's readings active in the time slice t and the set of contexts is represented by $\mathrm{Cn}_{\mathrm{t}} = (\mathrm{Cn}_{\mathrm{t}}^{1}, \ldots \mathrm{Cn}_{\mathrm{t}}^{\mathrm{n}})$ based on the application's environment. However, limiting the count of context variables for controlling CPT (Conditional Probability Table) size and, learning in the training phase.

DBN requires to define sensors and state transitions. The probability distribution $Pb(S_t|V_t)$ represents how sensor information is affected by the system's current state or the sensor model while its state transition model is $Pb(V_t|V_{t-1}, Cn_t)$, expresses the probability that a state variable has a certain value, given its previous value and current context. The DBN used is a first-order Markov model and the a specific system state in time slice t, i.e., v_t can be defined as Equation (1):

$$\mathrm{Bl}\ (\mathrm{v}_{\mathrm{t}}) = \mathrm{Pb}\ (\mathrm{v}_{\mathrm{t}}|\mathrm{S}_{1,\mathrm{t}}, \mathrm{Cn}_{1,\mathrm{t}})$$

Bayes Filter analogous procedure is followed for a practical formulation of *belief* and by applying Bayes rule it is possible to express Equation (1) as Equation (2)

$$\mathrm{Bl}\ (\mathrm{v}_{\mathrm{t}}) = \mathrm{Pb}(\mathrm{v}_{\mathrm{t}}|\mathrm{S}_{1:\mathrm{t}}, \mathrm{Cn}_{1:\mathrm{t}}) = \mathrm{Pb}(\mathrm{v}_{\mathrm{t}}|\mathrm{S}_{1:\mathrm{t}-1}, \mathrm{S}_{\mathrm{t}}, \mathrm{Cn}_{1:\mathrm{t}}) = \\ \eta.\mathrm{Pb}(\mathrm{S}_{\mathrm{t}}|\mathrm{v}_{\mathrm{t}}, \mathrm{S}_{1:\mathrm{t}-1}, \mathrm{Cn}_{1:\mathrm{t}}).\mathrm{Pb}(\mathrm{v}_{\mathrm{t}}|\mathrm{S}_{1:\mathrm{t}-1}, \mathrm{Cn}_{1:\mathrm{t}}) \ldots\ldots(2)$$

where η- Normalizing Constant. In a Markov assumption, the sensor nodes in $S_t$do not depend on context variables Cn_t, in a state variable S_t, and assuming sensor measurements are mutually independent, the value of a parent node S_t, can be expressed as Equation (3)

$$\mathrm{Pb}(\mathrm{S}_{\mathrm{t}}|\mathrm{v}_{\mathrm{t}}, \mathrm{S}_{1:\mathrm{t}-1}, \mathrm{Cn}_{1:\mathrm{t}}) = \mathrm{Pb}(\mathrm{S}_{\mathrm{t}}|\mathrm{v}_{\mathrm{t}}, \mathrm{Cn}_{1:\mathrm{t}}) = \mathrm{Pb}(\mathrm{S}_{\mathrm{t}}|\mathrm{v}_{\mathrm{t}}) = \prod_{\mathrm{s}_{\mathrm{t}}^{\mathrm{i}}} \mathrm{Pb}(\mathrm{s}_{\mathrm{t}}^{\mathrm{i}}|\mathrm{v}_{\mathrm{t}}) \ldots\ldots\ldots\ldots(3)$$

where s_t^i is the specific value of the sensor *i* in a time slice *t*. Moreover, the last term in Eq. (2) can also expressed as Equation (4):

$$\mathrm{Pb}(v_t|S_{1:t-1}, Cn_{1:t}) = \sum_{v_{t-1}} \mathrm{Pb}(v_t, v_{t-1}|S_{1:t-1}, Cn_{1:t}) = \alpha.\sum_{v_t-1} \mathrm{Pb}(v_t|v_{t-1}, S_{1:t-1}, Cn_{1:t}).\mathrm{Pb}(v_{t-1}|S_{1:t-1}, Cn_{1:t}) \text{....(4)}$$

where α - normalizing constant. Cn_t can be carefully omitted from the last term, since V_{t-1} does not depend on the next context Cn_t if the next state V_t is not considered. Thus, using the Markov assumptions on Equation (4) can be expressed as Equation (5):

$$\mathrm{Pb}(v_t|S_{1:t-1}, Cn_{1:t}) = \alpha.\sum_{v_{t-1}} \mathrm{Pb}(v_t|v_{t-1}, Cn_t).\mathrm{Pb}(v_{t-1}|S_{1:t-1}, Cn_{1:t-1}) = \alpha.\sum_{v_{t-1}} \mathrm{Pb}(v_t|v_{t-1}, Cn_t).\mathrm{Bl}(v_{t-1}) \text{....(5)}$$

by substituting Equation (3) and (5) in (2), belief can be defined with a recursive Equation (6)

$$\mathrm{Bl}\,(v_t) = \eta.\prod_{s_t^i} \mathrm{Pb}\left(s_t^i \middle| v_t\right).\sum_{v_{t-1}} \mathrm{Pb}(v_t|v_{t-1}, Cn_t).\mathrm{Bl}(v_{t-1}) \text{........(6)}$$

where α -integrated with normalization constant η. Using Equation (6), inference can be executed by storing only two slices of DBN where time and space updating network's belief are independent of the sequence's length. Computational complexity involved in Equation (6) is O(n + m), where n – no of sensors and m – no of possible values of V_t and overall complexity of $Bl(v_t)$ for all V_t is $O(m^2 + m \cdot n)$.

Convolution Neural Network based Prediction

This work uses CNN for knowledge mining and DF outcomes and has three layers namely input, convolution and softmax layers which are applied for reducing computational overheads and increasing prediction accuracy. CNN works in two parts where the first part generates deep features from raw data using convolutions while the second part connects features to an MLP for classification. They are detailed below

Input layer. This layer is denoted by N × k neurons, where k –input data no, N – data length.

Convolutional layer. This layer convolutes preceding layer's data using filters where the convolution filter's parameters are chosen based on domain knowledge or experiments. Examples of parameters are m – no of filters, s - convolution stride, r k × l – size of the filter with k as a variable data count of the previous layer and l the length of filter. f, a nonlinear transformation function acts on the layer. If preceding layer has k-variate data with N as data length then Convolutional operations results in m-variate data whose length can be $\left\lfloor \frac{N-l}{s} + 1 \right\rfloor$ and $\lfloor . \rfloor$ is rounding values.

Softmax layer: This layer uses a Softmax function to calculate distribution chances of an event in different events n. This function calculates every target text's occurrence in a target text. The calculated probabilities help in determining target text based on given inputs. The outputs probabilities lie in a range of 0 to 1. When the function is used in multi-classifications, it returns classes with high probabilities based on the target text. An exponential of the given input value and sum of all exponential values within inputs are computed. The ratio between input exponential value and sum of all exponential values is output by the function. If the output value of softmax function is high in multi-classifications it implies a better probability than other values. Though SoftMax determine multi-class probabilities, it has its own limits. The function becomes expensive as classes count increases. In such situations, candidate sampling is used for limiting SoftMax layer's scope to a specific set of classes.

Output layer. There are n neurons which correspond n feature classes in the output layer. Completely connected to the feature layer, it takes maximum output neurons as a class label for inputs in classifications. CNN trains using samples $((v_1,u_1),(v_2,u_2),\ldots,(\mathrm{v}_N,\mathrm{u}_N))$ where $v_t \in R^{N\times k}$, $u_t \in R^n$ for $1 \le t \le N$. High-order features $\boldsymbol{v}_t$ is the input while the vector $u_t$denotes output. Network trains based on the following steps:

Step 1 Network Initialization by finalizing input layer neurons count and output layer for classifications. Initialize bias and weights using a random number. Select η - learning rate, f - activation function and sigmoid function in Equation (7):

$$f(y) = sigmoid(y) = \frac{1}{1+e^{-y}} \quad (7)$$

Step 2 Randomly select a sample from the training set
Step 3 Compute each layer's output where Convolutional output can be written as Equation (8)

$$C_r(t) = f\left(\sum_{i=1}^{l}\sum_{j=1}^{k} y(i+s(t-1),j)\omega_r(i,j) + b(r)\right) \quad (8)$$

Where $y \in R^{N\times k}$ – input's higher order features/output of the previous layer, s - convolution stride, $C_r(t) =$ - t^{th} component of r^{th} feature map, ωr ∈Rl×k - weights and **b**(r) - bias of the r^{th} convolution filter.

(i) The outputs from the output layer can be written as Equation (9)

$$O(j) = f\left(\sum_{i=1}^{M} z(i)\omega_f(i,j) + b_f(j)\right), j = 1,2, \ldots, n \quad (9)$$

Where z - feature layer's final feature map, b_f - output layer's bias and $\omega_f \in R^{M\times n}$ is the connection of weights between the feature and output layers. Mean-square error is depicted in Equation (10)

$$E = \frac{1}{2}\sum_{k=1}^{n} e(k)^2 = \frac{1}{2}\sum_{k=1}^{n}(O(k) - y(k))^2 \quad (10)$$

Step 4 Update weight and bias using gradient descent method as in Equation (11)

$$\mathrm{p} = \mathrm{p} - \eta\frac{\partial \mathrm{E}}{\partial \mathrm{p}} \quad (11)$$

Where p – parameter value refers to ω_r, ω_f , b, or $b_f$in the proposed CNN

Step 5 Select a different training sample and go to Step 3; do this until all samples are trained.

Step 6 Increase iteration number by one for the next iteration. When = previously defined maximum value which exit the algorithm\ else Step 2.

Results and Discussion

RECALL: It is the total relevant instances retrieved where

$$\text{Recall} = \frac{|\{\text{relevant outcome}\}\cap\{\text{Predicted outcome}\}|}{|\{\text{relevant outcome}\}|}$$

PRECISION: Precision is defined as the ratio of correctly found positive observations to all of the expected positive observations and in this case retrieved relevant instances where

$$\text{Precision} = \frac{|\{\text{relevant outcome}\}\cap\{\text{Predicted outcome}\}|}{|\{\text{Predicted outcome}\}|}$$

F1 score is defined as the weighted average of Precision as well as Recall. As a result, it takes false positives and false negatives.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Accuracy is calculated in terms of positives and negatives as follows:

Accuracy = (TP+FP)/(TP+TN+FP+FN)

The proposed method was compared with existing DFA methodology. Table 1 lists comparative performances of the proposed and DFA methods.

Table 1 Comparative performances of CDFT

Metrics	Methods	
	DFA	CDFT
Accuracy	60	82
Precision	69	97
Recall	74	91
F-Measure	71.41	93.9

Figure 2 depicts comparative performances of the proposed CDFT and DFA methods in terms of Accuracy and Prediction performances.

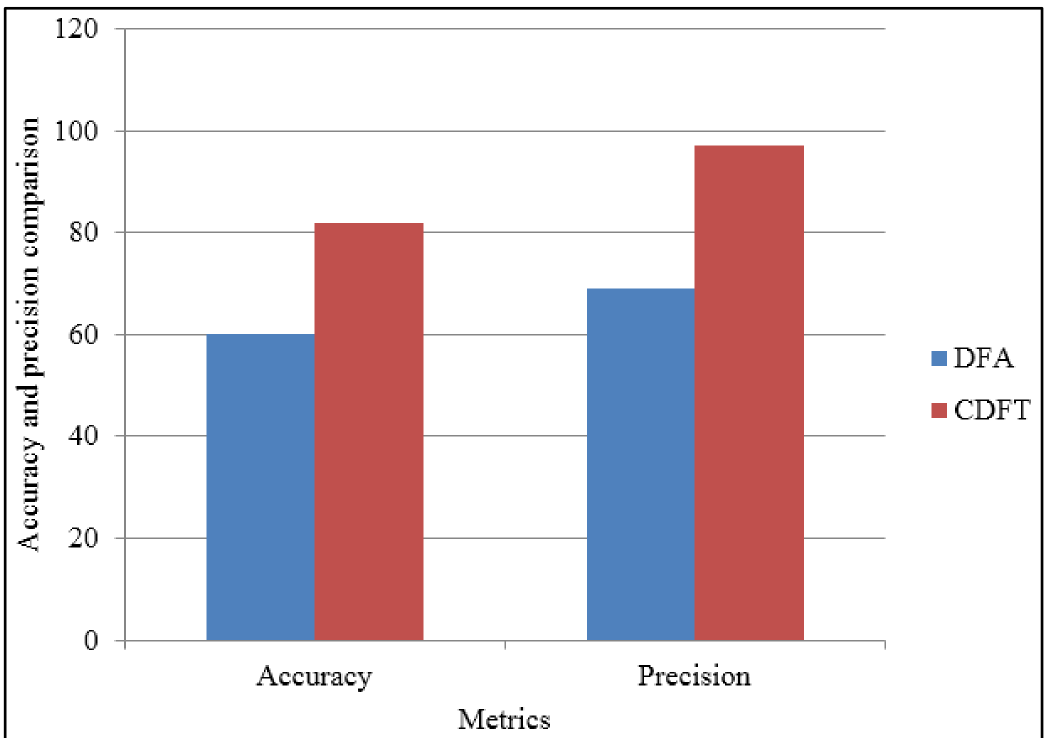

Fig. 2 Accuracy and Prediction performances of CDFT and DFA

The analysis from Figure 2 shows a better performance from CDFT. It shows an improvement of above 20% in accuracy and Precision when compared to DFA. Figure 3 depicts comparative performances of the proposed CDFT and DFA methods in terms of Recall and F1 measures.

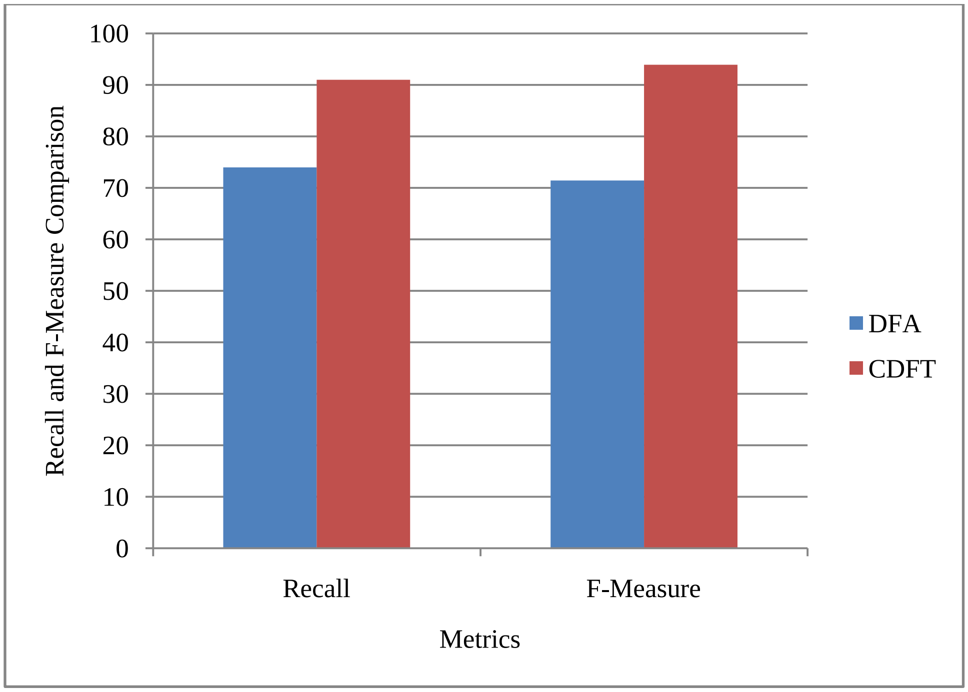

Fig. 3 Recall and F-Measure comparisons

From Figure 3 it evident that the proposed CDFT performs better than DFA as it shows an increase of 15% in recall values and 20% more in F1 Measure when compared to DFA technique.

Conclusion

Data management is a critical task in mammoth sensor and IoT generated data. Managing this huge volume of data, cleaning this data from impurities is an inherent need. DF processes combine data from multiple sources based on their similarity focussing on many factors. Many Data Fusion approaches have been proposed without detailing on the context for integrating data in fusion tasks. This work covers this gap in contexts with its proposed CDFT. The proposed system initially gathers data from IoT devices and pre-processes it to make it clean for fusion processing. In this work, boundary based noise reduction algorithm was used for data pre-processing in an attempt to label the unlabelled attributes in the gathered data for accurate data fusions. After pre-processing Context aware data fusion was performed for combining multiple IoT devices data together with a concern of contexts. Finally this combined data was learnt using the convolution neural network for data fusions performance was checked. The proposed work, CDFT was analyzed in a simulated matlab environment. The results proved that the proposed technique obtains optimal outcomes than existing techiques.

References

Luthra, S., Garg, D., Mangla, S.K., & Berwal, Y.P.S. (2018). Analyzing challenges to Internet of Things (IoT) adoption and diffusion: An Indian context. *Procedia Computer Science*, *125*, 733-739.

Agiwal, M., Saxena, N., & Roy, A. (2019). Towards connected living: 5G enabled Internet of Things (IoT). *IETE Technical Review*, *36*(2), 190-202.

Čolaković, A., & Hadžialić, M. (2018). Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, *144*, 17-39.

Lampropoulos, G., Siakas, K., & Anastasiadis, T. (2019). Internet of things in the context of industry 4.0: An overview. *International Journal of Entrepreneurial Knowledge*, *7*(1), 4-19.

Adeel, A., Gogate, M., Farooq, S., Ieracitano, C., Dashtipour, K., Larijani, H., & Hussain, A. (2019). A survey on the role of wireless sensor networks and IoT in disaster management. In*Geological disaster monitoring based on sensor networks,* 57-66. Springer, Singapore.

Prasad, R.C., Gandham, S.R., Yadav, N., Deen, K., Chang, S.C., Kulshreshtha, A., & Gupta, A. (2017).Automatically determining sensor location in a virtualized computing environment*U.S. Patent No. 9,733,973*. Washington, DC: U.S. Patent and Trademark Office.

Diaz, A., & Sanchez, P. (2016). Simulation of attacks for security in wireless sensor network. *Sensors*, *16*(11), 1932.

Alsayyari, A., Kostanic, I., Otero, C.E., & Aldosary, A. (2017). An empirical path loss model for wireless sensor network deployment in a dense tree environment. In *2017 IEEE Sensors Applications Symposium (SAS),* 1-6.

Manrique, J.A., Rueda-Rueda, J.S., & Portocarrero, J.M. (2016). Contrasting internet of things and wireless sensor network from a conceptual overview. In *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData),* 252-257.

Pramanik, P.K.D., Upadhyaya, B.K., Pal, S., & Pal, T. (2019). Internet of things, smart sensors, and pervasive systems: Enabling connected and pervasive healthcare. In *Healthcare data analytics and management*, 1-58.

Luis Bustamante, A., Patricio, M.A., & Molina, J.M. (2019). Thinger. io: An Open Source Platform for Deploying Data Fusion Applications in IoT Environments. *Sensors*, *19*(5), 1044.

Cross, E.S., Williams, L.R., Lewis, D.K., Magoon, G.R., Onasch, T.B., Kaminsky, M.L., & Jayne, J.T. (2017). Use of electrochemical sensors for measurement of air pollution: correcting interference response and validating measurements. *Atmospheric Measurement Techniques*, *10*(9), 3575.

Goffi, A., Bordogna, G., Stroppiana, D., Boschetti, M., & Brivio, P.A. (2020). A Scalable Synthesis of Multiple Models of Geo Big Data Interpretation. *Journal of Software Engineering and Applications*, *13*(6), 104-128.

King, R.C., Villeneuve, E., White, R.J., Sherratt, R.S., Holderbaum, W., & Harwin, W.S. (2017). Application of data fusion techniques and technologies for wearable health monitoring. *Medical Engineering & Physics*, *42*, 1-12.

Qi, J., Yang, P., Newcombe, L., Peng, X., Yang, Y., & Zhao, Z. (2020). An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure. *Information Fusion*, *55*, 269-280.

Berry, M.G., Stack, C.M., Magstadt, A.S., Ali, M.Y., Gaitonde, D.V., & Glauser, M.N. (2017). Low-dimensional and data fusion techniques applied to a supersonic multistream single expansion ramp nozzle. *Physical Review Fluids*, *2*(10), 100504.

Pfeffer, M., Uschmajew, A., Amaro, A., & Pfeffer, U. (2019). Data fusion techniques for the integration of multi-domain genomic data from uveal melanoma. *Cancers*, *11*(10), 1434.

Liu, Z., Meyendorf, N., & Mrad, N. (2018). The role of data fusion in predictive maintenance using digital twin. In *AIP Conference Proceedings*, AIP Publishing LLC, 1949(1), 020023.

Traini, S., Sciullo, L., Trotta, A., & Di Felice, M. (2019). Practical Indoor Localization via Smartphone Sensor Data Fusion Techniques: A Performance Study. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 1-7.

Martín-Morató, I., Cobos, M., & Ferri, F.J. (2017). Analysis of data fusion techniques for multi-microphone audio event detection in adverse environments. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, 1-6.

Singh, P., Pareek, V., & Ahlawat, A.K. (2017). Designing an energy efficient network using integration of KSOM, ANN and data fusion techniques. *International Journal of Communication Networks and Information Security*, *9*(3), 466-473.

Thupakula, K. (2017). *Data Fusion Techniques for Object Identification in Airport Environment* (No. 2017-01-2109). SAE Technical Paper.