PARTIAL MULTI-VIEW OUTLIER DETECTION BASED ON COLLECTIVE LEARNING

PRESENTATION BY

AJINKYA RAWANKAR - 2018201020

PRANAV VERMA - 2018201047

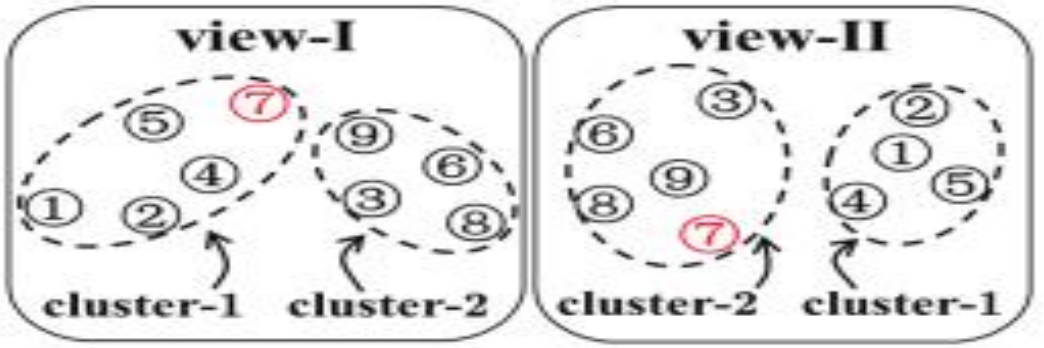
AMRIT KATARIA – 2018201067

PRANJAL PATIDAR - 2018201094

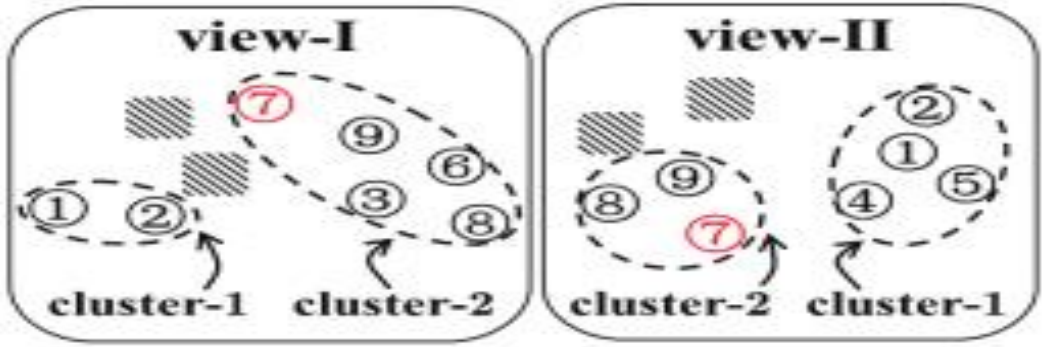
INTRODUCTION

- Data is collected from various domains, thus it contains multiple views.
- Data is not consistent ,that means it contains anomalies.
- Moreover data is not complete within each view.
- So this project aims at detecting the outliers in a multi-view data by first completing the data.

WHAT IS **MULTI-VIEW** DATA ?

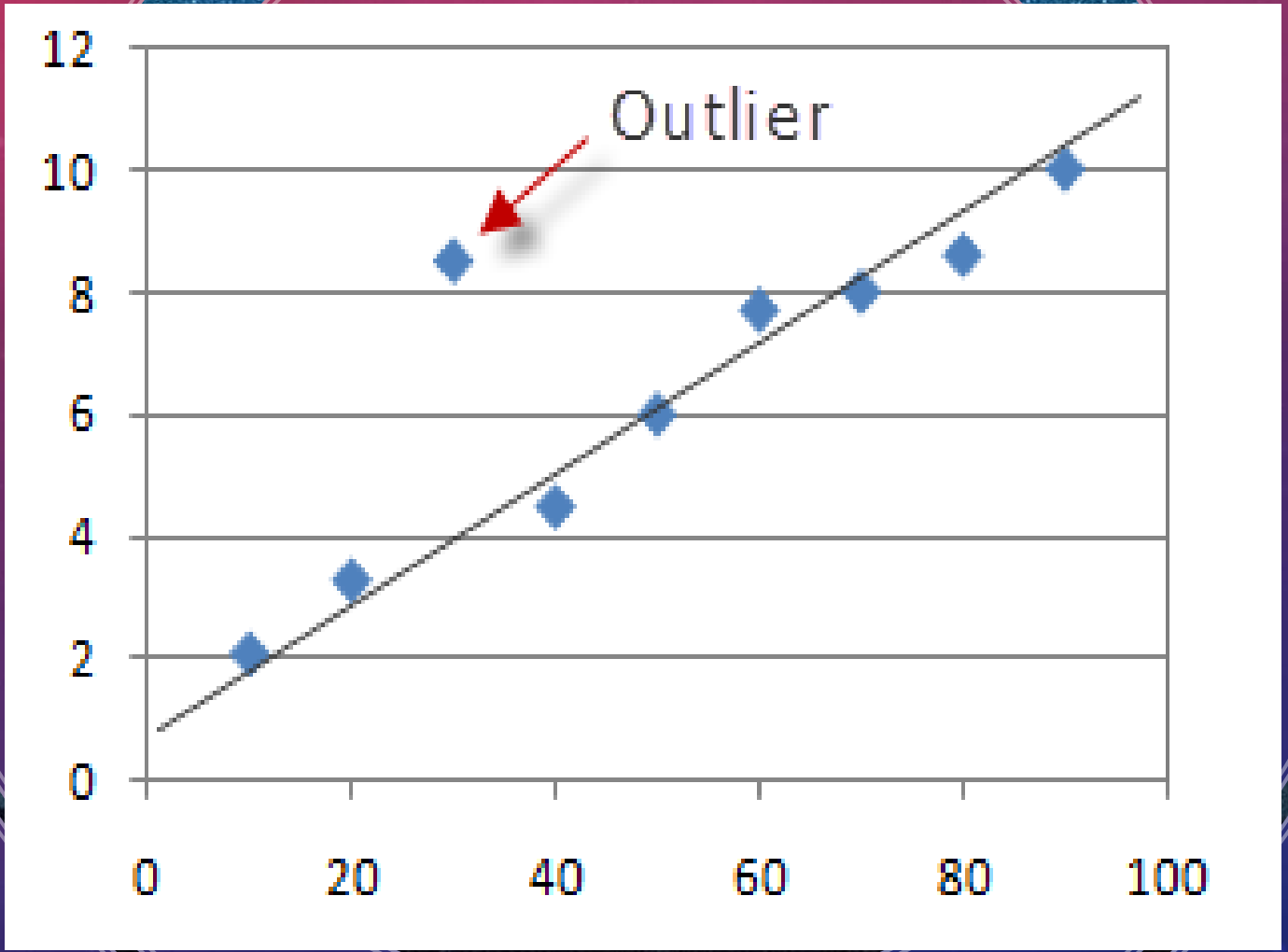


(a) Complete multi-view data



(b) Partial multi-view data

WHAT IS **OUTLIER** ?



WHAT IS COLLECTIVE LEARNING ?

WHAT IS REAL LIFE APPLICATION ?

- Intrusion detection
- Social media
- Health care

WHAT WE HAVE TO DO ?

- Generation of Partial Data
- Generation of Outliers
- Initialization missing data
- Recovering missing data
- Outlier detection

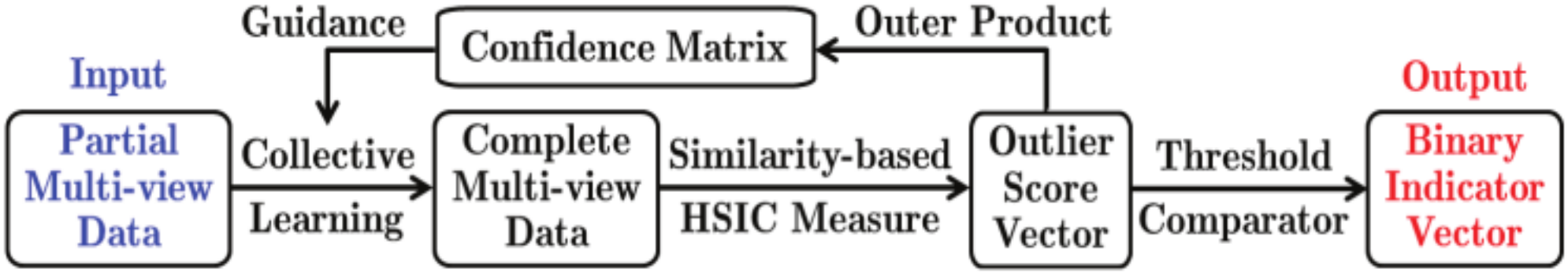


Figure 3: The flowchart of our proposed method to detect outliers for partial multi-view data.

- # GENERATION OF PARTIAL DATA

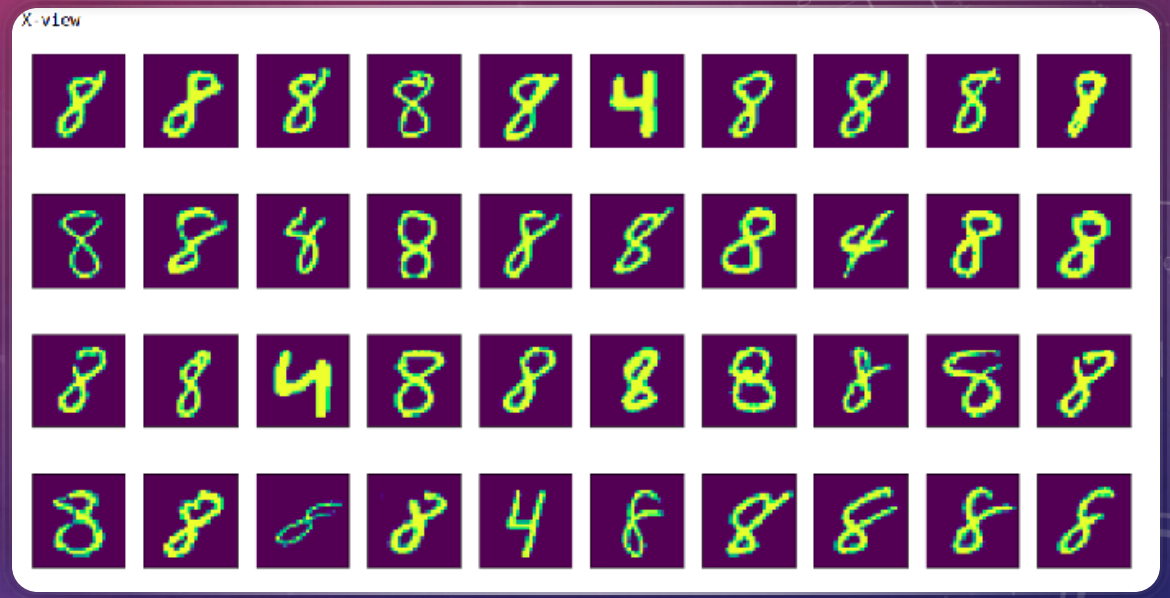
- 50 SAMPLES FROM EACH CLASS TAKEN.

- REMOVED FEW SAMPLES FROM EACH CLASS.

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^{n_c} \\ \mathbf{X}^{n_x} \\ \mathbf{X}^{n_y} =? \end{bmatrix} \text{ and } \mathbf{Y} = \begin{bmatrix} \mathbf{Y}^{n_c} \\ \mathbf{Y}^{n_x} =? \\ \mathbf{Y}^{n_y} \end{bmatrix},$$

GENERATION OF OUTLIERS

- SELECT FEW OF SAMPLES FROM 1 CLASS AND SWAP IT FROM ANOTHER CLASS.



- # INITIALIZATION OF MISSING DATA

  - INITIALIZE THE MISSING DATA WITH THE AVERAGE OF THE CLASS

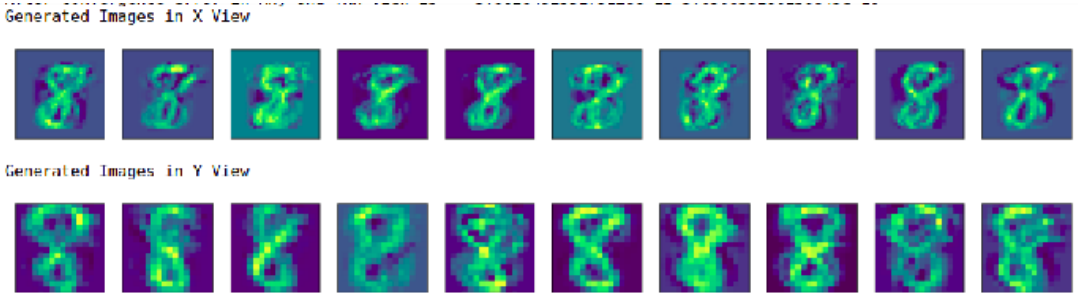
RECOVERING MISSING DATA

- RECOVERING DATA BASED ON COLLECTIVE LEARNING

$$\mathbf{P} = diag\,(\mathbf{C})\mathbf{H}\mathbf{X}\mathbf{X}^T\mathbf{H}.$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{n_c n_c} & \mathbf{P}^{n_c n_x} & \mathbf{P}^{n_c n_y} \\ \left(\mathbf{P}^{n_c n_x}\right)^T & \mathbf{P}^{n_x n_x} & \mathbf{P}^{n_x n_y} \\ \left(\mathbf{P}^{n_c n_y}\right)^T & \left(\mathbf{P}^{n_x n_y}\right)^T & \mathbf{P}^{n_y n_y} \end{bmatrix}.$$

$$\mathbf{Y}^{n_x} = -\left(\mathbf{P}^{n_x n_x}\right)^{-1}\left[\left(\mathbf{P}^{n_c n_x}\right)^T \mathbf{Y}^{n_c} + \mathbf{P}^{n_x n_y}\mathbf{Y}^{n_y}\right].$$



- # OUTLIER DETECTION

$$\mathbf{W}_{ij}^{(X)} = \begin{cases} 1\,, & j \in \mathcal{N}_i \text{ or } i \in \mathcal{N}_j \\ 0\,, & \text{otherwise} \end{cases}$$

- SIMILARITY MATRIX GENERATION USING KNN

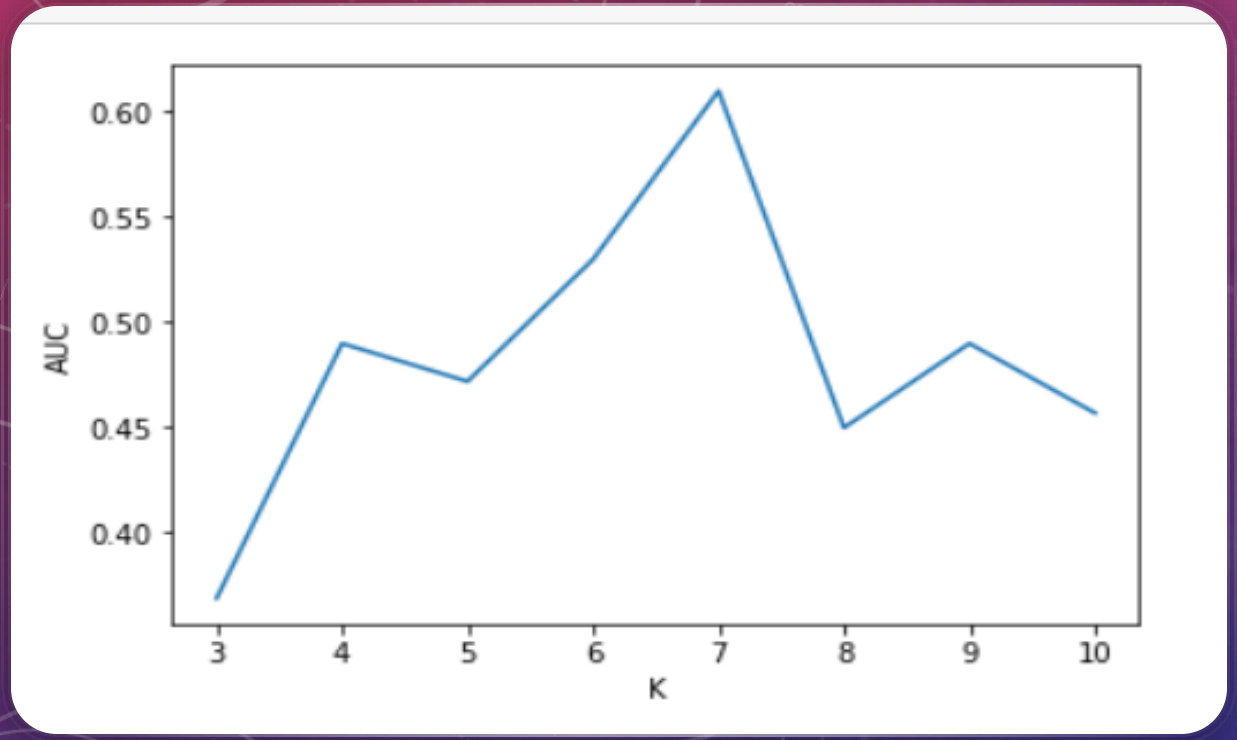
- OUTLIER SCORE VECTOR CREATION

$$\mathbf{s}\left(i\right) = \boldsymbol{\Delta}_{ii} \;\text{ with }\; \boldsymbol{\Delta} = \mathbf{H}\mathbf{W}^{(X)}\mathbf{H}\mathbf{W}^{(Y)}.$$

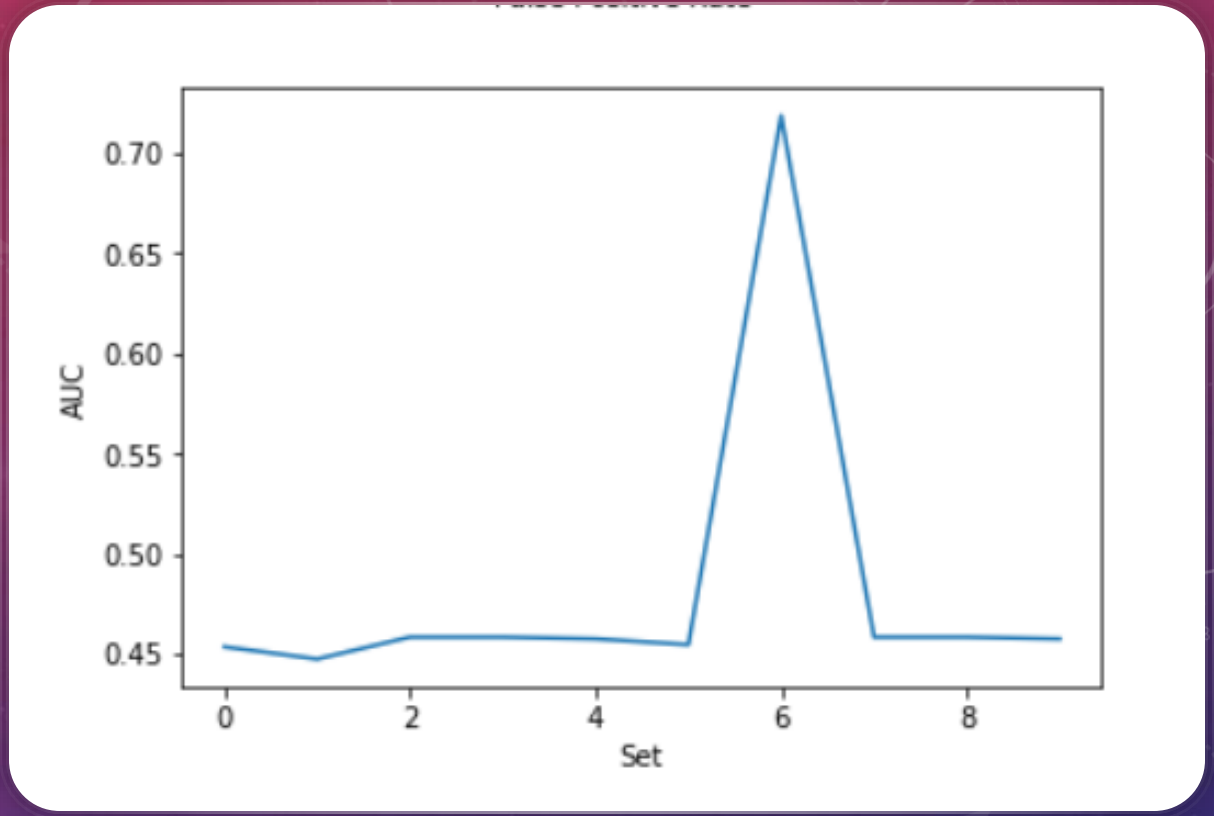
- DETERMINING THE OUTLIERS BY COMPARING SCORES WITH A CHOSEN THRESHOLD

$$\mathbf{s}\left(i\right) < \tau, \mathbf{o}\left(i\right) = 1; \text{otherwise}, \mathbf{o}\left(i\right) = 0.$$

RESULTS AND ANALYSIS



K vs AUC



Dataset vs AUC

THANK YOU!