3-10-2021

To Tolerate or To Impute Missing Values in V2X Communications Data?

Daoming Wan
University of Windsor

TO TOLERATE OR TO IMPUTE MISSING VALUES IN V2X COMMUNICATIONS DATA?

by

Daoming Wan

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2021

To Tolerate or To Impute Missing Values in V2X Communications Data?

by

Daoming Wan

APPROVED BY:

S. Samet
School of Computer Science

J. Wu
Department of Electrical and Computer Engineering

R. Razavi-Far, Co-Advisor
Department of Electrical and Computer Engineering

M. Saif, Co-Advisor
Department of Electrical and Computer Engineering

February 10, 2021

Declaration of Co-Authorship and Previous Publication

I. CO-AUTHORSHIP

I hereby declare that this thesis incorporates material that is the outcome of my research under the supervision of Dr. Mehrdad Saif and Dr. Roozbeh Razavi-Far. In general, the key ideas, methodology development, software programming, validation verification, data curation, writing, and visualization were performed by the author and the contribution of co-authors was primarily through the provision of study materials. Dr. Razavi-Far provided conceptualization ideas, methodology development, formal analysis, conducting research, resources, writing - review and editing, supervision, project management, and funding acquisition. Dr. Saif assisted investigation, conducting research, supervision, and funding acquisition. Dr. Mozafari helped writing - review and editing.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-authors to include the co-authored material in my thesis.

I certify that, with the above qualification, this thesis and the research to which it refers is the product of my own work.

II. PREVIOUS PUBLICATION

This thesis partly includes the original papers that have been previously submitted, to be submitted, or published in peer reviewed journals and conferences as provided in the following table.

Thesis Chapter	Publication title/full citation	Publication status
Chapters 1-6	D. Wan, R. Razavi-Far and M. Saif, “Cooperative Clustering Missing Data Imputation,” 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, 2020, pp. 1039-1045, doi: 10.1109/SMC42975.2020.9283484.	published
Chapters 1-6	D. Wan, R. Razavi-Far, M. Saif, N. Mozafari, “COLI: Collaborative Clustering Missing Data Imputation” 2021.	submitted
Chapters 1-6	D. Wan, R. Razavi-Far, M. Saif, N. Mozafari, “To Tolerate or To Impute Missing Values in V2X Communications Data?” 2021.	to be submitted

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

III. **General**

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Misbehavior detection is a critical task in vehicular ad hoc networks. However, state-of-the-art data-driven techniques for misbehavior detection are usually conducted through complete V2X communication data collected from simulated experiments. This thesis evaluates the main strategies for the treatment of missing values to find out the best match for misbehavior detection with incomplete V2X communication data. This thesis proposes three novel methods for imputing and tolerating missing data. The first two are novel imputation methods that are based on cooperative clustering and collaborative clustering. The latter is a missing-tolerant method that is an ensemble learning based on the random subspace selection and Dempster-Shafer fusion. The effectiveness of the proposed techniques is evaluated in the ground truth vehicular reference misbehavior data. Moreover, a multi-factor amputation framework has been developed to induce missingness over V2X communication data with different missing ratios, mechanisms, and distributions. This framework provides a comprehensive benchmark resembling missingness over V2X communication data. The proposed methods are compared with some missing-tolerant and imputation methods. The attained results over benchmark data are analyzed and indicated the winner treatments in each aspect.

Acknowledgements

Throughout the writing of this thesis, I have received a great deal of support and assistance. I would like to express my sincere appreciation to many people who encourage and help me through my research and study period.

First of all, I would like to express my sincere gratitude to my supervisors Dr. Mehrdad Saif and Dr. Roozbeh Razavi-Far from the Department of Electrical and Computer Engineering at the University of Windsor. They always guided me with useful instructions when I met some troubles and had questions about my research and writing. I am also appreciated for the fantastic opportunities they provided me and their continued suuport. They made me have an amazing experience in my Master's study, and I thank them wholeheartedly.

In addition, I would like to thank my thesis committee members, Dr. Jonathan Wu from the Electrical and Computer Engineering Department, and Dr. Saeed Samet from the School of Computer Science for taking the time out of their busy schedule to participate in my seminar, and I am gratefully indebted to their very valuable comments on this thesis. Their passionate participation and input enabled this research to be possible.

Finally, I must express my very profound gratitude to my parents for providing me with continual support and continuous encouragement throughout my years of study and through the period of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AWVRF	Adjusted Weight Voting Random Forest
CCI	Cooperative Clustering Imputation
COLI	Collaborative Clustering Imputation
DA	Data Augmentation
EMI	Expectation Maximization Imputation
FCM	Fuzzy c-means
FCMI	Fuzzy c-means Imputation
FS	Feature selection
GA	Genetic Algorithm
INSERT	Amputation method
KMI	k-means Imputation
MAR	Missing At Random
MCAR	Missing Completely At Random
MDS	Misbehavior Detection System
MI	Mean Imputation
MLP	Multilayer perceptron
MNAR	Missing Not At Random
MTE-RD	Missing-tolerant ensemble with RSM and Dempster-Shafer fusion.

NMI	Normalized Mutual Information
NRMSE	Normalized Root Mean Square Error
PAM	Partition Around Mediods
PAMI	Partition Around Mediods Imputation
RCEIFBC	Robust Clustering Ensemble based on an Iterative Fusion of Base Clusters
RSM	Random subspace
SkNNI	Sequential k-Nearest Neighbor Imputation
surrRF	Surrogate decision Random Forests
SVR	Support Vector Regression
V2X	Vehicle to Everything
VANET	Vehicle Ad-hoc Network
V-ELM	Voting based extreme learning machine

List of Symbols

b	Belief function
c	Cluster
d	Number of iteration
e	Number of import cluster algorithms
g	Number of missing patterns
h	Hidden layers for ELM
i	Loop indicator
j	Loop indicator
k	k nearest neighbors
l	Label indicator
m	Number of rows
n	Number of features
n_X	Number of missing features
nof	Number of features in random subspace
$o^u_{r,\mathcal{B}}$	Overlapped clusters between $R_\mathcal{B}$ and c^u_r
p^l_t	Probability prediction of C_t for class l
q	Number of subsets for INSERT
r	Clustering result indicator
$\hat{s}$	Missing sample for IMSERT amputation
t	Loop indicator
u	Cluster indicator
v	Voting for each cluster
w	Number of sub-clusters for CCI
$x_{i,j}$	Value of j-th feature in sample x_i
$\ddot{x}$	The indicator for missing cells

$\hat{x}_i$	Imputed sample
y	Feature value
z	Test sample
A	Clustering algorithms
B	Best result indicator
C_t	$t-th$ sub-classifier
$\mathcal{D}$	Probability distribution function
D_t	Probability distribution for RSM
F	Indices of selected features
K	Normalization constant
L	Number of classes
M	Membership degree for fcm
N	Number of samples in subset $\mathcal{X}^l(t)$
P_t	Probability prediction of C_t
R	Set of clustering results
$R_{\mathcal{B}}$	Best result for COLI
$\hat{R}_r$	Updated result r
T	Number of sub-classifiers
$\mathcal{U}$	Obtained imputation model
V	Temporary merged cluster for CCI
$\mathcal{V}$	Voting matrix
X_o	Original dataset
$X_{incomplete}$	Incomplete dataset for INSERT
X	Incomplete dataset for imputation
$\hat{X}$	Complete dataset
X_I	Incomplete subsets
X_C	Complete subsets
$\mathcal{X}(t)$	$t-th$ Training subset

$\hat{X}_I$	Imputed subsets
$\mathcal{Y}$	Feature sets
Y	Final prediction result
$\mathcal{Z}$	Matrix for indicators of missing cells
α	Significance level
β	Update factor
γ	Indices of available sub-classifiers
ε	Correlation coefficient
η	Number of clusters
θ	Decision profile
ι_j^k	Centroid of k-th cluster in R_j
κ_t	Weighted sum score for t-th subset
λ	Minimum distance cluster
μ	Support of each class
ν	Missing pattern for INSERT
ξ	Parameter for missingness
ρ	Missing ratio
σ	Probability of missingness
τ	Number of available clusters
ψ_j^k	Imputation quality of c_j^k
ω	Feature weights
Γ	Performance measure matrix for CCI
Θ	Decision template
$\hat{\Theta}$	Decision template for z
Φ	Proximity function
Ψ_j	Imputation quality of j-th partition
$\hat{\Psi}_r$	Imputation quality of updated partition
Ω	Symmetric matrix for CCI

Chapter 1

Introduction

Recently, the advancement of Internet of Things and wireless technology has led to a significant attention in the Vehicular Ad hoc Network (VANET). It builds a timely robust network to communicate between vehicle to everything (V2X). V2X exchanges basic safety messages through wireless communications to maintain traffic order, and, thus, plays an vital role in the intelligent transportation systems [1].

1.1 Motivation

Due to the highly dynamic nature of the system and frequent exchange of emergency and safety messages through wireless channels, VANET is highly exposed to attacks [2]. The malicious nodes may send fake or harmful information resulting in loss of revenue and life. Therefore, there is a need to detect misbehavior messages to guarantee the secure operation of V2X communications.

Most experiments for misbehavior detection systems (MDS) are usually conducted with complete simulated datasets [3]. However, VANET applications utilize the wireless sensor network to realize communication, and in reality received data may contain missing values for various reasons such as errors in sensor readings, synchronization, sensor faults, communication malfunctions, and malicious attacks. The presence of missing value always leads to inaccurate predictions and performance reduction of some classification algorithms [4, 5, 6, 7, 8, 9]. Therefore, handling missing values is a crucial task in misbehavior detection systems.

1.2 Contributions

The framework of this thesis enables an empirical comparison of the performance of the missing data treatment strategies with a triple objective. The former is to examine which imputation technique outperforms other imputation methods for misbehavior detection with incomplete V2X communications data. The second is to find out which missing-tolerant technique outperforms other missing-tolerant methods. The third one is to study the impacts of imputation and tolerant techniques in misbehavior detection performance. The performance of each strategy is evaluated by the publicly available V2X communications datasets in terms of accuracy, F-measure, and imputation quality for imputation methods. Meanwhile, the computational complexity of each strategy is also treated as a critical criterion to select the best method. The explorations of this work are practical and informative that incomplete environments are rarely discussed in V2X communications.

The incomplete environments are simulated by introducing a multi-factor amputation framework. It is proposed to create a set of benchmark data to evaluate the main strategies to treat missing values in V2X communications data for misbehavior detection purposes. Moreover, two novel methods for missing data imputation have been proposed to improve imputation quality. These two novel imputation methods apply state-of-the-art ensemble clustering techniques and collaborative frameworks to optimize the individual clustering-based imputation methods. A further contribution has been made for the proposed missing-tolerant method. It is a missing-tolerant ensemble, which can classify the incomplete sample directly. To study the efficiency of these three newly introduced methods, they are compared with several state-of-the-art missing data imputation and missing-tolerant methods over V2X communications data. More details of these novelties are illustrated in the following subsection.

1.3 Novelties

This work aims to evaluate the main strategies for the treatments of missing values to find out the best match for misbehavior detection with incomplete V2X communications data. Thus, a multi-factor amputation framework is proposed to simulate incomplete environments, and novel algorithms are proposed to achieve better performance, comparing to the existing treatments of missing values:

1.3.1 Amputation Methods

A multi-factor amputation framework, called INSERT, is introduced to generate multiple incomplete datasets from an original complete dataset. This amputation framework systematically induces missingness with different missing mechanisms, ratios, and distributions. It adopts and combines the ideas from previous amputation methods, making it more scientific and applicable.

1.3.2 Cooperative Clustering Imputation

The proposed imputation method is based on a cooperative clustering framework (CCI), whose consensus function is designed by means of a performance matrix. The clustering results obtained by various clustering algorithms generate a set of sub-clusters based on the similarity among the partitions. Then, the consensus function is utilized to decide which two sub-clusters should be merged. The final clustering model is attained when the merging procedure can not reach further improvement. The experimental results indicate that the proposed method has a better imputation performance than individual clustering-based imputation techniques.

1.3.3 Collaborative Clustering Imputation

The proposed collaborative clustering-based imputation (COLI) uses the imputation quality as key information to be exchanged between different clustering algorithms. COLI makes use of a confusion matrix enabling each clustering algorithm to split or re-cluster their partitioning. The experimental results indicate that COLI outperforms individual clustering-based imputation techniques and other commonly used imputation techniques.

1.3.4 Missing-Tolerant Ensemble

The proposed missing-tolerant method in this work, called MTE-RD, is a missing-tolerant ensemble, which uses the random subspace selection to train individual classifiers, and then combines their outcomes by means of the Dempster-Shafer fusion module. This method is the first time to apply the Dempster-Shafer fusion module to aggregate the predictions of the incomplete sample.

1.4 Outline

The rest of this study is organized as follows:

Chapter 2 initially describes the traditional treatments for missing values. Then, two main strategies, imputation, and missing-tolerant are introduced. After that, statistical-based and machine learning-based imputation methods are briefly clarified. Finally, four state-of-the-art missing-tolerant methods, which are used as competitors, are also introduced.

Chapter 3 explains a novel imputation method, called CCI. The development procedures of CCI are then proposed in this chapter. In order to simulate the incomplete environments, the principle and the development procedures of the multi-factor amputation methods are introduced to generate missingness for the V2X communications datasets. Then, the details of the V2X communications datasets used in this work are explained. Finally, the imputation results of this method with other clustering-based imputation methods are also included in this chapter.

Chapter 4 proposes another novel imputation method, called COLI. This novel imputation method is compared with other imputation algorithms in fifteen publicly available datasets. The experimental results of the proposed method and its competitors are also presented in this chapter.

Chapter 5 presents the development procedures of the proposed missing-tolerant method, called MTE-RD. Then, twenty V2X communications datasets are selected to evaluate the performance of imputation and missing-tolerant methods in terms of accuracy and F-measure. Finally, the significance test and the computational complexity of the proposed methods and their competitors are also illustrated.

Chapter 6 concludes this thesis. It gives the answer of the best match for V2X communications datasets with incomplete environments.

Chapter 2

Previous Works

In order to deal with incomplete data, one common method is to discard incomplete samples. This method works well with a few incomplete samples; nevertheless, it results in the loss of important information. A more practical approach to process missing values is imputation. The imputation methods fill a missing value with one or a set of estimations. They have been widely applied in various research fields [10]. Another class of approaches to treat missing data is those classification methods that can tolerate missing values and directly classify incomplete samples. Many missing-tolerant approaches have been successfully applied for incomplete data classification [11, 12, 13, 14]. The rest of this section explains some state-of-the-art imputation methods and missing-tolerant methods in detail.

2.1 Imputation

The imputation approaches are mainly classified into two types, namely, statistical and machine learning-based techniques [15]. Among the statistical approaches, the mean imputation (MI) is the simplest statistical imputation method, which replaces the missing values of a target feature with the mean of that feature [16].

Expectation maximization imputation (EMI) [17] is an iterative imputation meth -od with two steps. In the expectation step, observed values are used to compute the mean and covariance matrix, which estimate the missing values. In the maximization step, the mean and covariance are updated based on the imputed values. These steps are repeated until the mean and covariance matrix are stabilized. Data augmentation imputation (DA) is an iterative method to infer unknown parameters and imputes missing values in a stochastic manner [18]. DA makes use of EM for initial estimation of parameters and randomly imputes missing values by means of these parameters. It then uses the imputed data to obtain a Bayesian posterior distribution and updates the parameters. DA repeats these steps in a Markov chain manner until convergence.

Machine learning algorithms are also widely used to process incomplete datasets. They explore the similarity between different features or different samples to obtain more reasonable estimations [19]. k-nearest neighbors imputation replaces missing scores of an incomplete target sample from a set of donors that are the k nearest neighbors of the target sample. Sequential k-nearest neighbor imputation (SkNNI) replaces missing values according to the order of missing rate [20]. Once an incomplete sample has been filled, it can be used as a donor for imputing the rest of the incomplete samples.

Clustering, as an unsupervised learning method, has been widely used to handle incomplete data. Some of these approaches combined other machine learning techniques to achieve better accuracy for missing values estimation.

k-means clustering [21] is popular for cluster analysis because of its simplicity. It randomly assigns k initial cluster centers and then assigns those samples that are closest to one centroid to that cluster. Each cluster updates its center by calculating the mean of its constituent samples. The iterations will end when the cluster centers are fixed [22]. As for the k-means imputation (KMI), the original dataset is divided into two complete and incomplete subsets. A k-means clustering model is first created based on the complete subset. The incomplete samples find their closest cluster center based on their available values, and their missing values are replaced with the corresponding values in their closest cluster centers [23].

Moreover, in [24], class labels have also been used for imputing missing values. In [25], the missing values are not merely estimated by means of its closest cluster centers as donors. It then refines the estimated values by using the multilayer perceptron

(MLP). This hybrid approach shows that it has more accurate estimations compared to normal k-means imputation.

In the fuzzy c-means (FCM) clustering, each sample belongs to more than one cluster. It uses membership degrees to indicate to what degree each sample belongs to each cluster. The fuzzy c-means imputation (FCMI) has two main steps. The first step is to create an FCM model with the whole dataset and obtain the membership degree for each centroid. The second step is to impute the incomplete sample as follows:

$$x_{i,j} = \sum_{t=1}^{\eta} M(x_i, \iota^t) * \iota^t(j) \tag{2.1}$$

where x_i represents the target incomplete sample, in which the $j - th$ features is missing, η presents total number of clusters, ι^t is the centroids of the $t - th$ cluster, and M is the membership degree of a sample x_i to the certain cluster ι^t.

Various strategies combine FCM with other machine learning algorithms for missing values imputation. In [26], Genetic Algorithm (GA) is applied to optimize the membership degree and the centroids of the FCM model. In [27], MLP refines the estimation values that obtained by FCMI. In [28], the parameters of FCM are optimized by Support Vector Regression (SVR) and GA. These hybrid approaches significantly improved the imputation quality, comparing to FCMI.

Partition around medoids (PAM) clustering has the same strategy with k-means to form partitions. In contrast to k-means, it chooses a representative medoid for each cluster. The chosen medoid is minimizing the sum of the distance between medoid and the samples in the same cluster [29]. To compare with k-means, PAM is robust to outliers. Partition around medoids imputation (PAMI) has a similar procedure with KMI. The missing values of an incomplete sample are replaced with the corresponding values from the nearest medoid.

2.2 Missing-Tolerant Methods

The missing-tolerant classification method is a kind of missing values treatment that can classify incomplete samples directly. They are mainly classified into two categories based on how subsets are created.

The first category creates complete subsets based on the missing pattern of the original dataset. Voting based extreme learning machine (V-ELM) [12] is also an advanced ELM algorithm. It can clarify incomplete data, where the standard ELM classifier cannot directly classify incomplete data. It splits the incomplete data into complete subsets based on Missing Patterns. Each complete subset is used to train on the ELM sub-classifier, and the weighted majority vote combines the output of these sub-classifiers. The weighted function is associated with the mutual information between the feature set and the class labels on the complete data subsets. V-ELM is an efficient method to handle incomplete data.

Another alternative approach has been developed in [30], which uses missing patterns and feature selection (FS) to create an ensemble for classification of incomplete data. On the one hand, the feature selection method can reduce the dimension of the feature space and, thus, decrease the running time. On the other hand, it significantly increases the chance that the ensemble correctly classifies incomplete samples. Hereafter, this method is called FS due to its focus on the feature selection process in this thesis.

Another category creates complete subsets based on the random subspace of the feature set. Adjusted Weight Voting Random Forest (AWVRF) [14], based on the Random Forest algorithm, classifies incomplete data without imputation. The predicted class will be preserved at the internal node if the primary splitting feature and the surrogate feature of that node are missing. The weight function will be adjusted by the influence of the missing value on the decision tree. Furthermore, the algorithm uses the weighted majority vote for making final decisions. It has been proved that the method outperforms other surrogate decision Random Forests (surrRF).

Another method is Learn^{++}.MF [13], which makes use of the random subspace method (RSM) to address the issue of incomplete data classification. It generates a group of random feature subsets for original data and creates several complete subsets based on these feature subsets. Then, it trains base classifiers with the complete subsets. In the test phase, an incomplete sample is merely predicted by those classifiers that are trained with those subsets whose features are not missing in the target sample. The outputs of base classifiers are combined through the majority of votes.

2.3 Summary

This chapter explains some treatments of missing values, including imputation methods and missing-tolerant methods. The imputation methods are introduced for statistical methods and machine learning-based methods. Especially, the clustering-based imputation methods are presented in detail. Then, four state-of-the-art missing tolerant methods are also introduced based on different strategies to select feature sets. These treatments are used as competitors to compare with the proposed methods.

Chapter 3

Cooperative Clustering Imputation

The main objective of this chapter is to develop a novel method for imputing missing values of incomplete datasets. The basic concepts of the cooperative clustering framework and the development procedures of the proposed cooperative clustering imputation are presented in detail. Then, In order to perform a proper verification of the proposed method, various incomplete general datasets and V2X communication datasets are required whose original values are available. Therefore, several general datasets of UCI depository and Vehicular Reference Misbehavior datasets (VeReMi) that are complete and publicly available are selected, and, then, missingness is induced on them to generate various incomplete datasets. The process of inducing missingness is called INSERT amputation. This chapter formally introduces the VeReMi dataset and a set of amputation strategies that systematically induce missingness over original datasets. Finally, the imputation quality of the proposed method is compared with individual clustering-based imputation methods in publicly available datasets and V2X communication datasets.

3.1 Cooperative Clustering

Cooperative clustering, also known as ensemble clustering, is a framework that aggregates a set of various clustering results to attain a better partition result than the individual clustering method. Different clustering algorithms, or the same clustering algorithm with different parameters, are used to partition a given dataset. Then, local partition results are combined by means of a consensus function. The consensus function explores the agreements between different local results and aggregates them to reach stable and robust final partitions. In recent years, many cooperative clustering methods have been published. The main diversity of these methods is in the design of the consensus function.

A robust clustering ensemble based on an iterative fusion of base clusters (RCEIFBC) has been proposed in [31] that has a two-step consensus function. It first combines sub-clusters that have a higher cluster-cluster similarity and then assigns the samples to combined clusters according to the sample-cluster similarity. Ayad and Kamel [32] applied a cumulative voting method as the consensus function to aggregate a set of clustering results. They described several cumulative vote-weighting schemes for computing empirical distributions summarizing the ensemble.

Kashef and Kamel [29] presented a cooperative model to combine several clustering algorithms with the same number of clusters. It generates a set of sub-clusters based on the cluster membership degree of each object and then merges those sub-clusters with the highest similarity. These pair-wise similarities of sub-clusters are presented by means of similarity histograms. The merging phase finishes when the preset number of clusters has been reached.

This work applies cooperative clustering to treat missing values of incomplete datasets, named CCI.

3.2 CCI Framework

The proposed method utilizes the cooperative clustering framework to aggregate multiple clustering algorithms. Then, the final consensus clustering model is used to impute missing values of an incomplete dataset. The following subsections present all steps in detail.

3.2.1 Initializing Clustering Algorithms

The incomplete dataset (X) is divided into two subsets: (1) complete subset (X_C) and (2) incomplete subset (X_I). The complete subset only contains those samples that have no missing values. The incomplete subset is used to examine the imputation performance. $A = \{A_1, A_2, \ldots, A_e\}$ presents e clustering algorithms. Each clustering algorithm generates a set of clusters based on X_C. Let $R = \{R_1, R_2, \ldots, R_e\}$ be the clustering results of all clustering algorithms. It should be noted that unlike other cooperative clustering frameworks, the proposed method does not require the imported clustering algorithms have the same number of clusters. Thus, the number of clusters in each R_i can be different.

3.2.2 Generating Sub-clusters

The cooperative phase is performed based on the association of various clustering approaches. Thus, before the cooperative phase, a set of disjoint sub-clusters ($S = \{S_1, S_2, \ldots, S_w\}$) are generated to present the agreement among these clustering results. w stands for the total number of sub-clusters. The sub-clusters are examined to find the overlapping samples between different clustering results. The samples are assigned to one sub-cluster, if and only if they always belong to the same cluster according to all clustering results. Let $\eta = \{\eta_1, \eta_2, \ldots, \eta_e\}$ be the number of clusters returned by each clustering algorithm.

In order to find the intersection of clustering results, the membership degree of each sample x is calculated. $mem(x)|_{R_i} \in \{1, 2, \ldots, \eta_i\}$ indicates sample x belongs to which cluster. Equation 3.1 defines the membership degree function:

$$mem(x) = \sum_{i=1}^{e} mem(x)|_{R_i} * (\eta_i)^{i-1} \tag{3.1}$$

If the membership degree of two samples x_1 and $x_2 \in X_C$ are the same, then $mem(x_1) = mem(x_2)$. These two samples belong to the same sub-cluster.

3.2.3 Cooperative Phase

In the cooperative phase, a new set of sub-clusters (V) are used to present the temporary merged results. For example, $V_{i,j}$ indicates that S_i and S_j sub-clusters are merged. These temporary merged results are stored in a symmetric matrix Ω. The number of elements in Ω matrix are $(w \times (w-1)/2)$.

$$\Omega = \begin{pmatrix} 0 & V_{1,2} & V_{1,3} & \dots & V_{1,w} \\ \vdots & 0 & V_{2,3} & \dots & V_{2,w} \\ \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & & V_{(w-1),w} \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

The purpose of merging sub-clusters is to find a clustering model that is most suitable for missing data imputation. In each iteration, only two sub-clusters are merged. Many cooperative clustering methods utilize normalized mutual information (NMI) or similarities as the criterion to merge sub-clusters. The pair-wise sub-clusters with the highest similarities or NMI are merged to achieve a better partition quality. However, since the better partition quality cannot guarantee more accurate estimations of missing values, another criterion is introduced to merge sub-clusters for the sake of enhancing imputation accuracy. Normalized Root Mean Square Error (NRMSE) is commonly used to evaluate the imputation performance. The equation of NRMSE is shown as follow:

$$NRMSE = \frac{\left\| \hat{X} - X_o \right\|_F}{\left\| X_o \right\|_F} \tag{3.2}$$

where $\hat{X}$ is the imputed data, X_o is the original complete data, and $\|.\|_F$ stands for the Frobenius norm.

The algorithm makes use of each temporary merged sub-cluster (V) to impute missing values of X_I. It takes each sample of the incomplete subsets and finds its nearest centroid of sub-clusters, and replaces missing values with corresponding feature values of the nearest centroid. The imputation performance (NRMSE) of each temporary merged result (V) are stored in a performance measurement matrix (Γ):

$$\Gamma = \begin{pmatrix} 0 & \psi_{1,2} & \psi_{1,3} & \dots & \psi_{1,w} \\ \vdots & 0 & \psi_{2,3} & \dots & \psi_{2,w} \\ \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & & \psi_{(w-1),w} \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

where $\psi_{i,j}$ present the imputation performance based on the temporary merged results $V_{i,j}$. A lower NRMSE indicates a better imputation performance. Thus, searching the minimum value in the performance matrix and finding its corresponding temporary merged result is the best merged solution for this iteration. For example, if $\psi_{3,4}$ is the lowest value in Γ, $V_{3,4}$ is the best merged solution for this iteration.

3.2.4 Consensus

At the end of each iteration, the best temporary merged result is stored in Z, which consists of a new set of sub-clusters, ω stores the performance measure of Z. For example, Z_i and Ψ_i present the best merged result and its performance measure in the $i - th$ iteration. If $\Psi_i < \Psi_{i-1}$, the iteration can continue. Otherwise, Z_{i-1} is considered as the final clustering model, which is used to impute the missing values of the incomplete subset X_I. The pseudo-code of the proposed CCI technique is presented in Algorithm 1.

Algorithm 1: Cooperative Clustering Imputation.

```
Input: $X$ and $A$
Output: $\hat{X}$
Definitions:
$X$: incomplete dataset,
$\hat{X}$: imputed dataset,
$A$: a set of clustering algorithms,
$R$: clustering results,
$S$: a set of sub-clusters,
$Z_i$: best temporary merged result in $i-th$ iteration,
$\Psi_i$: performance measure of $Z_i$,
$\Omega$: temporary cooperative result matrix,
$\Gamma$: performance measure matrix,
begin
    Split $X$ into $X_C$ and $X_I$ subsets
    $R \leftarrow A(X_C)$ /* $A$ partition $X_C$ into $R$ */
    $S \leftarrow mem(R)$ /* $R$ generate sub-clusters $S$ */
    $Z_0 = S$ /* Initialize best clustering results with the first sub-cluster $s$ */
    $\Psi_0 \leftarrow$ PERFORMANCE($S$)
    $\Omega \leftarrow$ TEMPORARY MERGE($S$)
    $\Psi_1 \leftarrow$ BEST PERFORMANCE($\Omega$)
    $i = 1$
    while $\Psi_i < \Psi_{i-1}$ do
        $i = i + 1$
        $\Omega \leftarrow$ TEMPORARY MERGE($Z_{i-1}$)
        $\Gamma \leftarrow$ PERFORMANCE MEASURE($\Omega$)
        $\Psi_i \leftarrow Min(\Gamma)$
        $Z_i \leftarrow$ BEST($\Omega$) /* find the best corresponding merged results in $\Omega$ */
    end
    $\hat{X}_I \leftarrow$ IMPUTE($Z_{i-1}, X_I$) /* missing values in $X_I$ are imputed based on
     final decision clustering model $Z_{i-1}$ */
    $\hat{X} = \hat{X}_I \bigcup X_C$
end
```

3.3 Datasets

The experiments, which are used to evaluate the proposed method, are performed with public datasets and V2X communication datasets. The details of public datasets are listed in Table 3.1.

Table 3.1 – Characteristics of the used datasets.

Datasets	No. of samples	No. of features
Iris	150	4
Wine	178	13
Glass	214	9
Statlog Heart	270	13
BUPA liver disorders	345	6
Breast Cancer Wisconsin	683	9
Pima Indians Diabetes	768	8
Dermatology	358	34
Difficult Doughnut	400	12
Yeast	1484	8

VeReMi is the public labeled V2X communications dataset, which contains simulated traffic behavior data, including representative samples of various attackers [33]. These datasets contain five different attackers (see Table 3.2), three traffic densities, and different attacker densities. The features of these datasets include transmission time, sender, attacker type, message ID, and actual position/speed vectors.

Table 3.2 – Attacker parameter

ID	**type**	**parameter**
2	Constant	x = 5560,y = 5820
3	Constant offset	Δx = 250, Δy = -150
4	Random	uniformly random in playground
5	Random offset	Δx,y uniformly between -300 and 300
6	Eventual stop	stop probability += 0.025 each update

These ground truth VeReMi datasets and publicly available datasets are not naturally incomplete. Thus, the amputation methods should be introduced to create several incomplete datasets for experiments.

3.4 Missing Data Amputation

Missing data are inevitable in V2X communications data. Although real V2X communications data contain missing values, there is a need to have a set of benchmark V2X communications data, where missing values are systematically induced over data. For this purpose, the ground truth VeReMi datasets that are originally complete are considered into account. Then, missingness is induced over each VeReMi dataset in various ways by means of a missing data amputation technique to generate multiple incomplete V2X datasets.

Missing values are very common for the V2X communications data. Incomplete messages are frequent according to the missing at random (MAR) mechanism. For instance, features about the weather usually have a great impact on the transmission strength of the signal, resulting in missingness over the received messages [34]. Although MAR is a very common reason for missingness in V2X data, the missingness might have other reasons. Thus, we aim to study the impact of other missing mechanisms in misbehavior detection. Therefore, for the sake of a more comprehensive study, there is a need to induce missing values over V2X communications data with various missing ratios, mechanisms, and distributions.

In this work, a multi-factor amputation framework is proposed, which generates missingness with different missing ratios, mechanisms, and distributions (INSERT). This section formally presents different components of the INSERT amputation framework that systematically induce missingness over VeReMi datasets and other publicly available datasets. INSERT induces missingness to each complete data with four different missing ratios (1%, 5%, 10%, and 20%), five probability distribution functions (Normal, Logistic, Exponential, Weibull, and Inverse Gaussian), and three missing mechanisms (MCAR, MAR, and MNAR). The probability distribution functions help to select a sample for inducing missingness. INSERT calculates weighted sum scores for all samples. These are all fitted on a probability distribution to compute the missing probability for each sample. Then, the probability random selection is used to select samples for inducing missingness.

3.4.1 MCAR Amputation

Let us assume X is a dataset of dimension of $m \times n$, in which x_{ij} stands for the cell on $i-th$ row and $j-th$ column of X, x_i stands for the $i-th$ sample of X, and $\mathcal{Y}_j$ stands for the $j-th$ feature of X. X can be split into two subsets $X = X_{obs} \cup X_{miss}$, where X_{obs} and X_{miss} are subsets of observed and missing samples of X. X_{obs} and X_{miss} include all values x_{ij}, where $\ddot{x}_{ij}$ are zeros and ones, respectively. Moreover, $\mathcal{Z}$ is a binary matrix of missing indicator of X that can be defined as $\mathcal{Z}_{m\times n} = \{\ddot{x}_{ij}\}_{i,j=1}^{m,n}$, in which $\ddot{x}_{ij}$ is one if x_{ij} is missing and is zero if x_{ij} is observed.

The probability distribution of $\mathcal{Z}$ might depend on X_{obs} and X_{miss}, which yield to describe the missingness mechanisms $\sigma(\mathcal{Z}|X,\xi)$ with a set of parameters ξ [35]. MCAR mechanism is completely unrelated to data and the missingness probability merely depends on a set of parameters ξ as follows: $\sigma(\mathcal{Z}|X,\xi) = \sigma(\mathcal{Z}|\xi)$. In other words, in MCAR (see Algorithm 2), the missing values are randomly scattered through data.

Algorithm 2: MCAR Amputation

```
Input:
Complete dataset: $X_o$
Missing ratio : $\rho$
Output: Incomplete dataset: $X_{incomplete}$
begin
    for $i \in [1, \rho mn]$ do
        $X_o[random(1,m), random(1,n)]$ = 'NaN'
    end
    $X_{incomplete} = X_o$
end
```

3.4.2 MAR Amputation

To simulate the missing at random (MAR) mechanism (see Algorithm 3), INSERT randomly divides the original dataset X into several disjoint sample subsets $X = \{X_1 \cup X_2 \cup \ldots \cup X_q\}$, in which q is the number of subsets, and each sample merely appears in one subset. The size of each subset is a user-defined parameter. For each subset, only one missing pattern ν can be applied to generate missing values. A missing pattern indicates which features must contain missing values. All missing patterns consider the same number of features to induce missingness. This number is determined based on the missing ratio.

The missingness is established by the principle of the MAR missing mechanism: $\sigma(\mathcal{Z}|X, \xi) = \sigma(\mathcal{Z}|X_{obs}, \xi)$, where the probability of missingness depends on ξ and observed values X_{obs}, i.e., causative features [36]. Then, a feature has missing values due to its observed values or another causative feature [17]. In this work, the correlation is considered into account to identify the systematic relationship between the missing feature and its causative feature. This mechanism firstly locates a pair of features with the largest correlation. It then sets one of these features as a causative feature and induces missing values on the other feature. If the missing pattern contains more than one missing feature, it then selects another pair of features, which has the second-largest correlation to induce missingness. It repeats these steps until the number of missing features has been reached. This process is repeated for each subset to reach the required missing pattern.

The next step is to decide which samples have to be selected to induce missing values for each missing pattern. Each feature has a weight ($\omega_i \in \omega$) that is determined by the user according to its importance. If the importance of features can not be found, more weights are assigned to the causative features. The weighted sum scores κ_i aggregates the impact of features in each sample $\kappa = \omega_1 \cdot y_1 + \omega_2 \cdot y_2 + \cdots + \omega_n \cdot y_n$.

The probability distribution function is then applied to the weighted sum scores to allocate the probability of being missing. In Figure 3.1, for instance, by applying

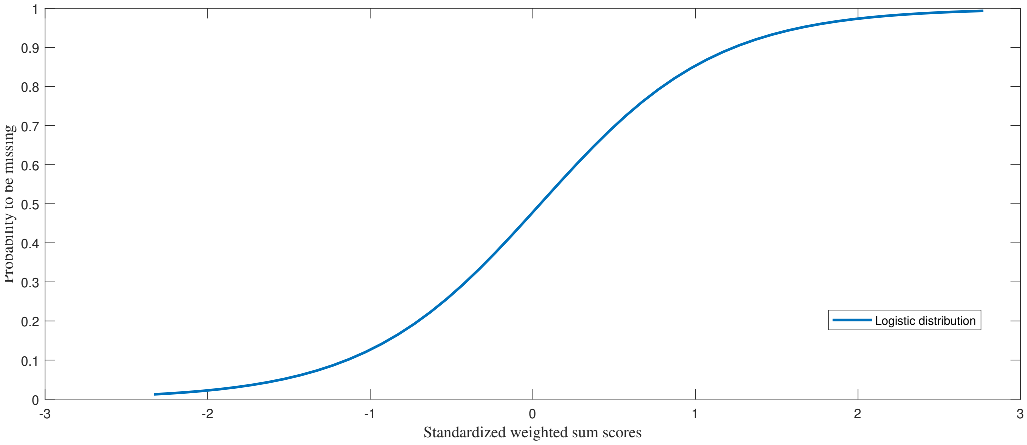


Figure 3.1 – Cumulative probability function of logistic distribution.

the logistic distribution on κ, the sample with a higher κ has a higher probability of being missing. Each sample in the subset is allocated the probabilities of being missing. The probability random selection is then used to determine missing samples until the number of missing samples has been reached. Finally, the missing values are induced in missing features and missing samples for each missing pattern.

3.4.3 MNAR Amputation

To simulate the missing not at random (MNAR) mechanism (see Algorithm 3), the probability of missingness not only depends on the observed values X_{obs}, but also depends on missing values X_{miss} as follows: $\sigma(\mathcal{Z}|X, \xi) = \sigma(\mathcal{Z}|X_{obs}, X_{miss}, \xi)$. INSERT applies a similar strategy to MAR. However, the process of determining the missing pattern and weight of each feature is different. The correlation cannot be used to determine missing patterns. In this case, each feature has the same opportunity to have missing values. The missing patterns are designed by the random selection of features to induce missingness until a predetermined number has been reached. The weighted sum score is based on the same equation. However, higher weights are assigned to the missing features. The probability distribution function is similarly used to select missing samples.

Algorithm 3: MAR and MNAR Amputations

```
Input:
Complete dataset: $X_o$
Missing ratio: $\rho$
Number of subsets (number of missing patterns): $q$
Distribution type: $\mathcal{D}$
Features weights: $\omega$
Output: Incomplete dataset: $X_{incomplete}$
Definitions:
$\nu_t^{MAR}$: Missing pattern for t-th subset in MAR.
$\nu_t^{MNAR}$: Missing pattern for t-th subset in MNAR.
$n_X$: Number of missing features for each subset.
$\kappa^t$: Weighted sum score for subset t.
$X^t(i,j)$: Value of i-th row and j-th column in subset of t.
$\sigma^t$: Probability of missingness for subset t.
$\hat{s_t}$: Missing samples for subset t.
begin
    /*Split $X_o$ into $q$ subsets */ $X_o = \{X_1 \cup X_2 \cup \ldots \cup X_q\}$
    $n_X = \lfloor \rho \cdot n/q \rfloor$
    for $t \in [1, q]$ do
        /* Compute correlation matrix of features */
        $\varepsilon = cor(X_t)$
        /* Find missing pattern for each subset */
        for $i \in [1, n_X]$ do
            /* Find indexes of most correlated features */
            $[\mathcal{Y}_1, \mathcal{Y}_2] = \arg\max_{i,j} \varepsilon(i,j)$
            $\nu_t^{MAR}(i) = \mathcal{Y}_1$
            $\varepsilon(\mathcal{Y}_1, \mathcal{Y}_2) = 0$
        end
        /* Design missing pattern based on random selection of $n_X$ columns for each
          subset in MNAR */
        $\nu_t^{MNAR} \leftarrow rand(X_t, n_X)$
        /* Create weights for each subset based on $\nu_t$ */
        if $\omega^t == 0$ then
            $\omega^t \leftarrow rand(\nu_t)$
        end
        for $i \in [1, m/q]$ do
            /* Get the $\kappa$ for each row in each subset*/ $\kappa^t(i) = \sum_{j \in \nu_t} \omega^t(j) * X^t(i,j)$
        end
        /* Get the probability of being missing for each row based on the distribution
          method */ $\sigma^t \leftarrow \mathcal{D}(\kappa_t)$
        /*Select missing values randomly based on $\sigma_m^t$ */ $\hat{s_t} \leftarrow rand(X_t \mid \sigma^t)$
        /*Create missing value for each subset */ $X_t(\hat{s_t}, \nu_t) =$ ‘NaN‘
    end
    $X_{incomplete} = \{X_1 \cup X_2 \cup \ldots \cup X_q\}$
end
```

Therefore, the INSERT amputation method is used to generate a set of incomplete datasets from each dataset. The set of incomplete datasets consist of three missing mechanisms (MCAR, MAR, and MNAR), four missing rates (1%, 5%, 10%, and 20%), and five distributions (Exponential, Normal, Logistic, Weibull and InverseGaussian). It should be noticed that the distributions are only applied to MAR and MNAR. Thus, the total number of incomplete datasets for each complete dataset is 44 (MCAR only has four incomplete datasets, MAR and MNAR both have 20 incomplete datasets).

3.5 Experimental Results

In this section, the experimental setting to evaluate the proposed method and its competitors is firstly explained. The attained results are analyzed and compared, which demonstrate that the cooperative clustering imputation improves the imputation accuracy, compared to individual clustering imputation methods. The proposed imputation technique and its competitors are evaluated in estimating missing values of the vehicle-to-everything (V2X) communication data [37].

3.5.1 Experimental Setting

The proposed cooperative clustering imputation approach is compared with three standard clustering-based imputation techniques, including KMI, FCMI, and PAMI. The parameters of these three imputation techniques are properly tuned. CCI makes use of three individual clustering algorithms, including *k*-means, fuzzy c-means, and partition around medoids. The parameters of these clustering algorithms are the same as KMI, FCMI, and PAMI. Therefore, if the imputation quality of CCI is better than the other three imputation techniques, it can prove that CCI is a more effective technique.

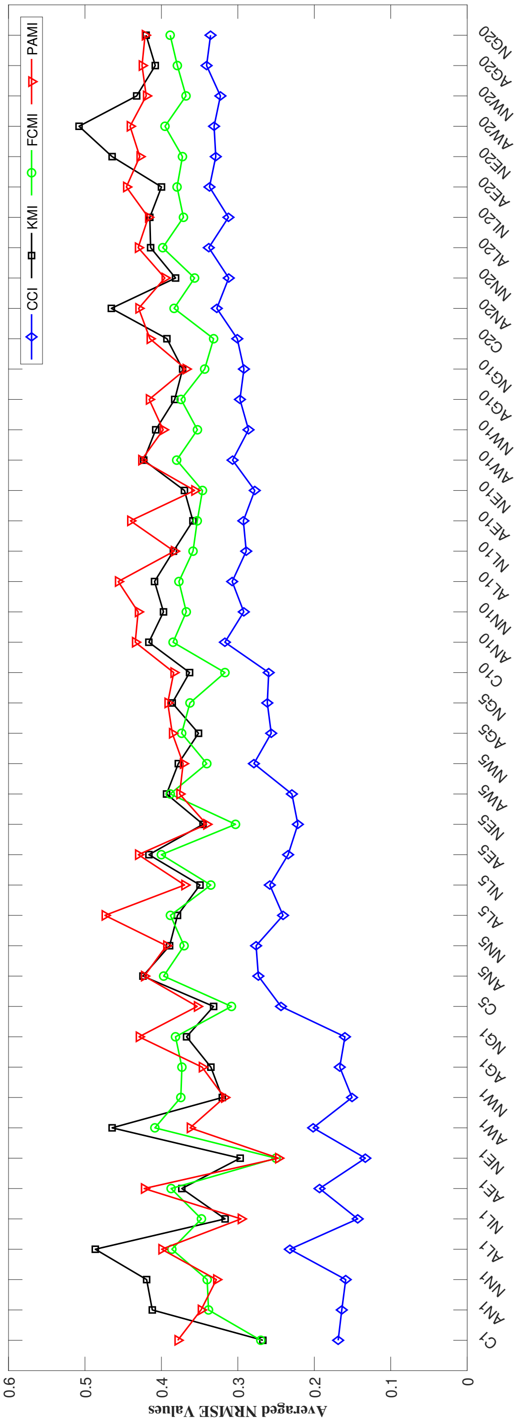


Figure 3.2 – The averaged NRMSE values over all incomplete datasets obtained by each imputation method for each combination of the missing mechanism, missing rates, and missing distribution.

3.5.2 Results Analysis

Figure 3.2 illustrates the averaged NRMSE values over all incomplete datasets obtained by each imputation method. The x-stick presents the combination of missing mechanism, missing rates, and missing distribution. The first letter means the missing mechanism, C stands for MCAR, A stands for MAR, and N stands for MNAR. This setting is also used in Figure 3.4. The second letter means missing distribution for MAR and MNAR, N stands for the normal distribution, L stands for the logistic distribution, E stands for the exponential distribution, W stands for the Weibull distribution, and G stands for the inverse Gaussian distribution. The last number stands for missing rate.

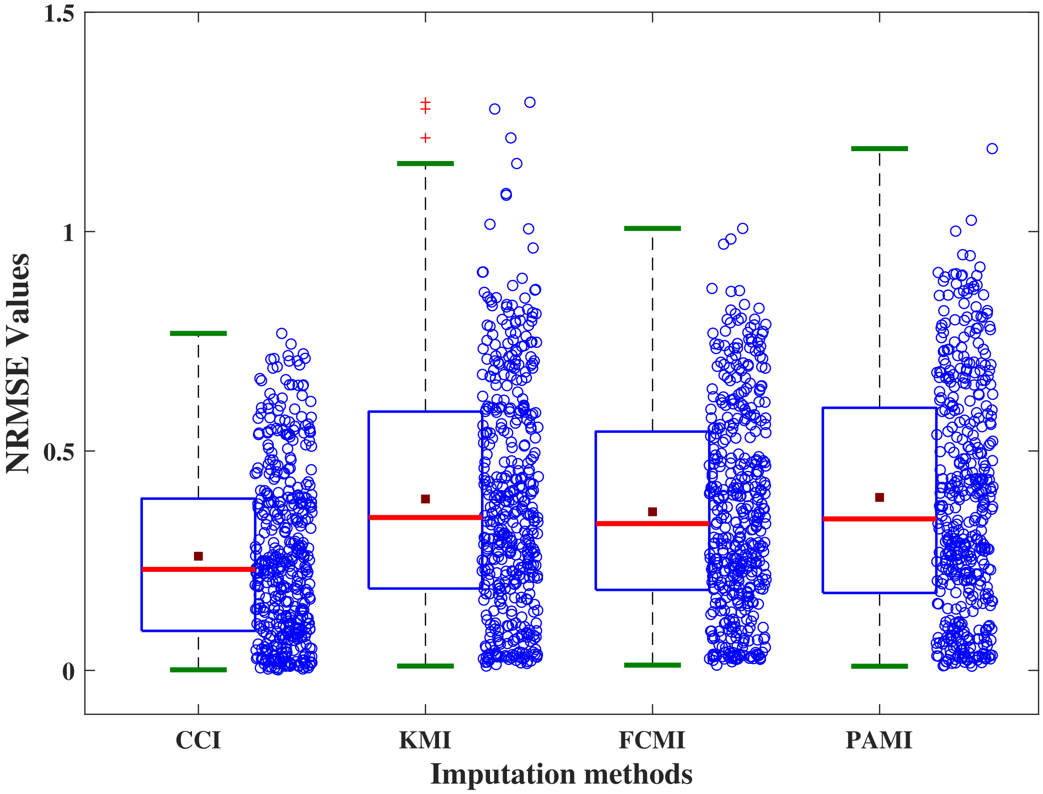


Figure 3.3 – Distribution of NRMSE values attained by each imputation method. The red crosses stand for outliers and the solid squares denote the average value for each box. The solid dashes stand for 1st and 99th percentiles. The red solid line is the median value for each method.

For example, AN1 means MAR missing mechanism, 1% missing rates with the normal missing distribution. A more accurate imputation has the lower NRMSE value. In this figure, the proposed method has a lowest averaged NRMSE values, which indicates that it outperforms the other three imputation techniques in all missing combinations.

Figure 3.3 presents the distribution of the attained imputation performance in terms of NRMSE overall incomplete datasets by each imputation method. The proposed imputation method has the lowest average and smallest box, which demonstrates that the proposed method has the best and steadiest imputation performance among other competitors. KMI has the largest variation, which indicates it has the most unstable performance. FCMI has a better imputation result comparing with KMI and PAMI.

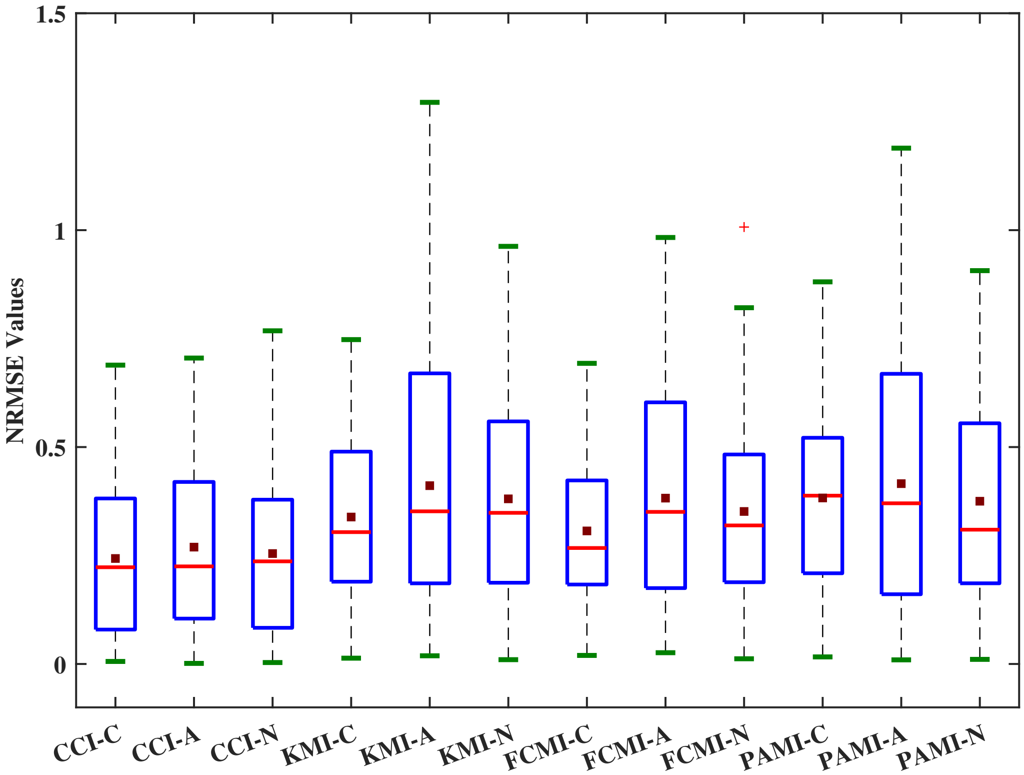


Figure 3.4 – Distribution of NRMSE values obtained by each method through all incomplete datasets with different missing mechanisms.

Figure 3.4 shows the distribution of imputation performance attained by each method through incomplete datasets with different missing mechanisms. In the figure, x-axis, the letters before the dash line present the imputation method, and the letter after the dash line shows the missing mechanism. For example, KMI-A stands for the imputation performance attained by k-means imputation for all incomplete datasets with MAR missing mechanism. To compare with competitors, CCI outperforms other imputation approaches in every missing mechanism.

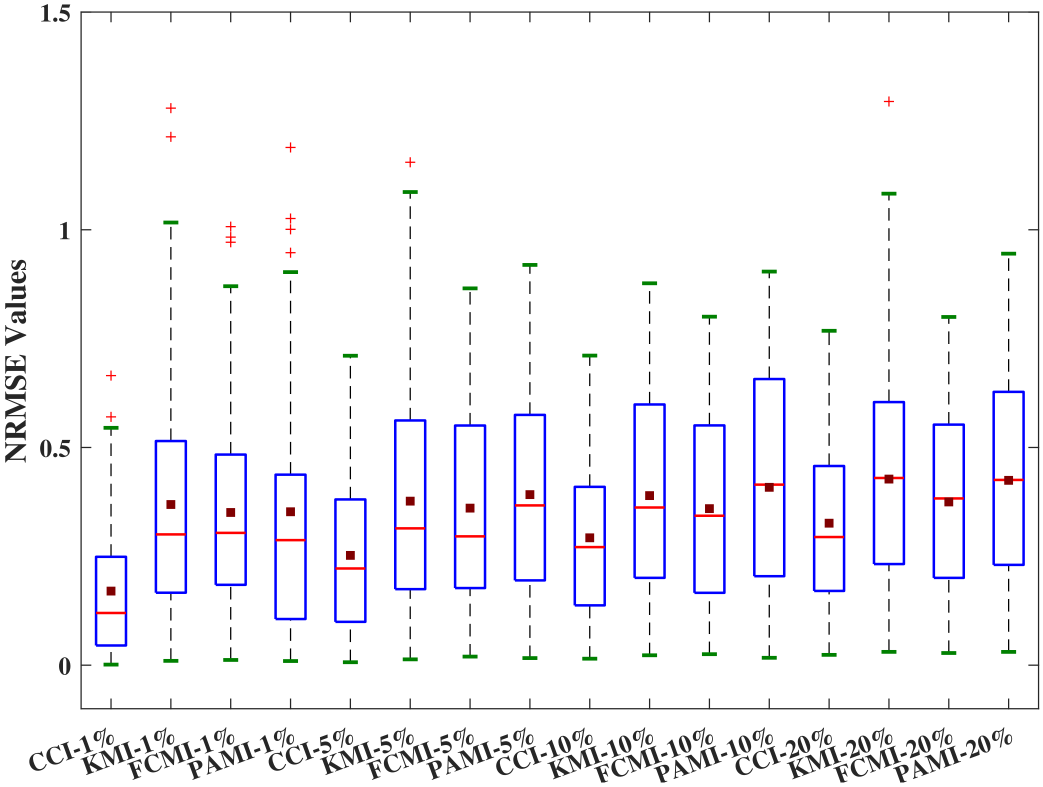


Figure 3.5 – Distribution of NRMSE values obtained by each method through all incomplete datasets with different missing rates.

Figure 3.5 illustrates the distribution of the imputation accuracies attained by each method through all incomplete datasets with different missing rates. With the increase of the missing rate, the imputation performance is decreased. The higher missing rates mean less complete samples that can be used for partitioning. Therefore, the clustering model contains a limited number of samples, and the incomplete sample only has a few options to use for estimation of its missing values. In this situation, the proposed method still has the best and stablest imputation performance for each missing rate.

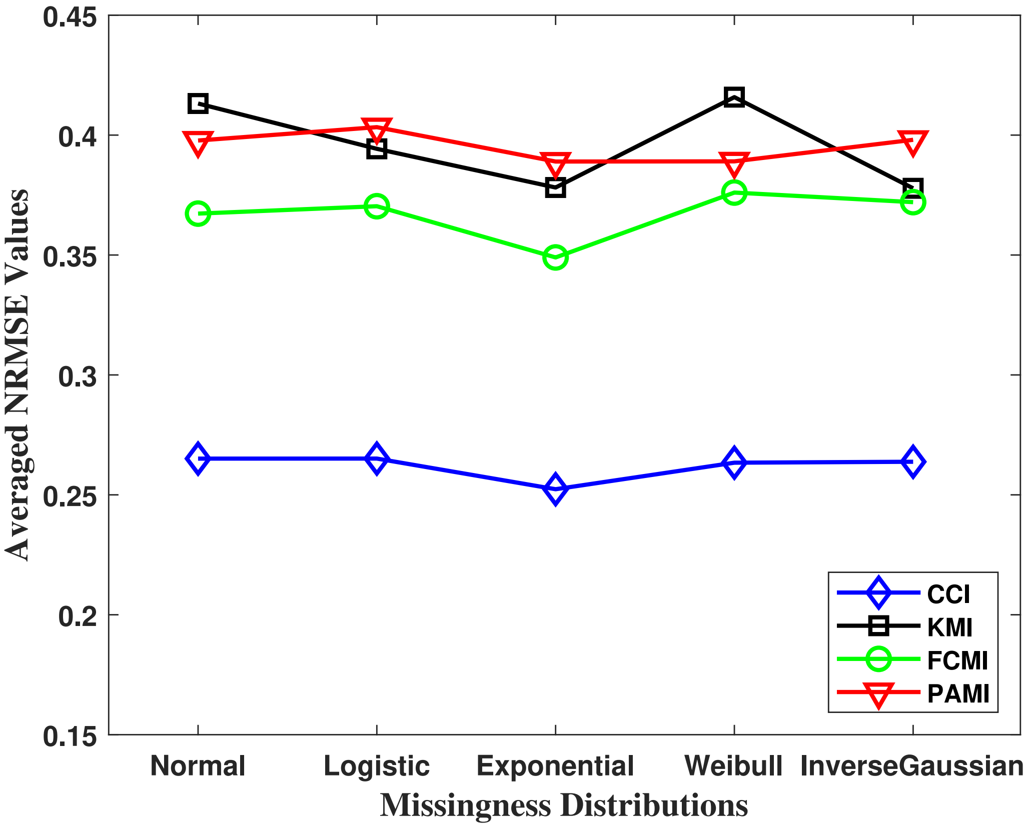


Figure 3.6 – Imputation performance of each method for different missing distribution.

Figure 3.6 presents the influence of the distribution in the generation of missing scores. In general, all imputation methods perform well on those incomplete datasets with the exponential distribution of the missing scores, comparing to other distributions. These clustering-based imputation methods are more suitable for the incomplete datasets with exponential distribution, and the performance of FCMI is always better than KMI and PAMI. However, the proposed method is the most stable technique overall missing distributions and always outperforms its competitors.

3.5.3 Analysing Results on Incomplete V2X Communication Data

Five datasets are selected from publicly available V2X communication datasets and used in our experiments. They are collected from different simulated scenarios. The details of these datasets are listed in Table 3.3.

Table 3.3 – Characteristics of the selected *VeReMi* datasets.

Datasets	No. of samples	No. of features
Scenario 1	1143	7
Scenario 2	1138	7
Scenario 3	1141	7
Scenario 4	1090	7
Scenario 5	1090	7

The INSERT amputation method is applied to generate 44 incomplete datasets with different missing mechanisms, rates, and distributions for each scenario. This results in generation of 220 incomplete VeReMi datasets. These incomplete datasets are used to evaluate the imputation performance of the proposed method and its competitors.

Figure 3.7 presents the imputation performance in terms of NRMSE attained by each imputation method overall incomplete V2X communication datasets. The proposed method, CCI, has the lowest average and median NRMSE values, which indicate that the proposed method is the most accurate method compared to other competitors. Moreover, CCI has the smallest box that illustrates the proposed method is the most stable method among other competitors through all V2X communication scenarios. The attained results show that the proposed method is the most efficient and stable technique to estimate missing values of the V2X communication data.

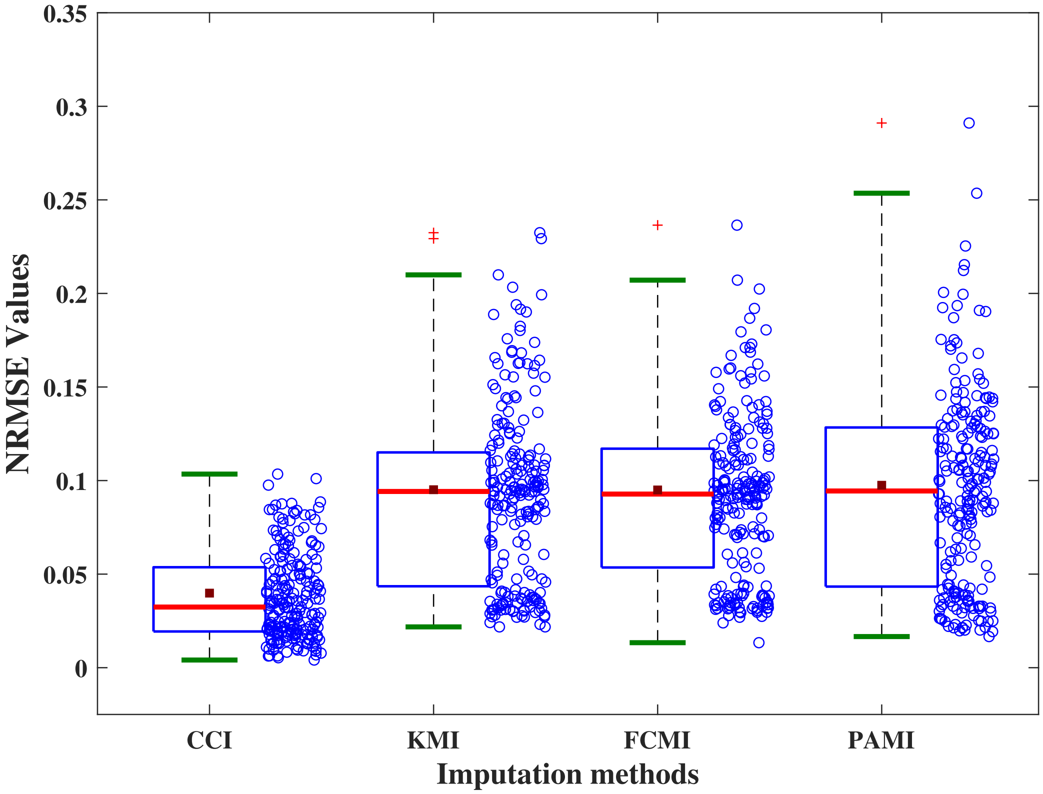


Figure 3.7 – Distribution of the NRMSE values attained by each imputation method in estimating missing values of incomplete VANET datasets. The red crosses stand for outliers and the solid squares denote the average value of each box. The solid dashes stand for 1st and 99th percentiles. The red solid line is the median of the NRMSE values attained by each method.

3.6 Summary

In this chapter, a novel imputation algorithm is proposed that is based on cooperative clustering. This chapter also develops a multi-factor amputation framework, called INSERT, to induce missingness over V2X data with various ratios, mechanism, and distributions, and create multiple incomplete benchmark data. For cooperative clustering imputation, various clustering algorithms generate a set of sub-clusters based on the agreement among themselves. Then, a consensus function is used to merge these sub-clusters to explore the best model for imputation. The proposed method is compared with three standard clustering-based imputation methods for imputing missing values of incomplete datasets, in which the missing scores are generated over ten public datasets and five V2X communication data with different missing mechanisms, missing rates, and missing distributions. The experimental results indicate that the proposed method significantly outperforms the competitors. In the future, CCI will be evaluated to impute missing values of more V2X incomplete datasets.

Chapter 4

Collaborative Clustering Imputation

This chapter illustrates another proposed imputation method COLI. The basic concepts of collaborative clustering are presented at first. Then, the development procedures of the proposed method are introduced in detail. It explains the exchange of information, which is the key factor in the development of the collaborative step. Later, two operators are considered into account to update the partitioning. Then, it presents the proposed imputation algorithm and the main steps. Finally, the experimental results of the proposed method and its competitors are presented.

4.1 Collaborative Clustering

Collaborative clustering is a framework in which different clustering results collaborate to reach a better partitioning on common data. Each partitioning refines itself by exchanging information with other partitionings. These partitionings can be obtained through different clustering algorithms or different parameters of the same algorithm. Moreover, these clustering results can be the partition of one data or the partition of subsets of data [38].

Collaborative clustering mainly has five critical characteristics: (1) partitioning information is being exchanged between different partitioning results, (2) different algorithms must have common characters, (3) the criteria are set to evaluate the performance of each partitioning, (4) different algorithms have a common goal to improve the consensus, (5) various operations are used to update each partitioning.

The implements of collaborative clustering are various. Some methods are devoted to creating a suitable framework for different clustering algorithms. A collaborative clustering has been developed in [39] to increase the agreements of partitioning among different clustering algorithms. This general method is applicable to all clustering algorithms. A similar collaborative clustering framework [40] is adopted for multiple consensus clustering, multi-view clustering, and alternative clustering. It iteratively exchanges information to obtain a consensus or an alternative clustering solution. This method is independent of the types and parameters of the clustering algorithms. Another general collaborative clustering method has been developed in [41] that is able to exchange information between various partitioning results obtained through different subsets of data.

Some collaborative clustering approaches only focus on a single clustering algorithm. In [42], the FCM clustering is applied to several independent subsets of data to obtain partitioning results. Then, these results exchange information about local partition matrices to acquire a common partition. Another collaborative clustering method is based on the k-means algorithm [43]. It improves the validity of the k-means algorithm through large distributed data. In order to accelerate the convergence, this method introduces a collaborative seeding among different partitioning results.

4.2 COLI Framework

Recently, a lot of work has focused on the use of multiple clusterings to improve the clustering process. However, there is little research done on missing value imputation using multiple clustering results. Moreover, existing approaches are based on Ensemble clustering, which combines multiple clustering results into an alternative approach without any collaboration between different partitions [44, 45]. This chapter proposes a novel imputation technique that is based on collaborative clustering. The fundamental concept of collaborative clustering is that the algorithms operate locally and then collaborate by exchanging information about their structure to improve their result. The proposed collaborative clustering-based imputation (COLI) treats the imputation quality as the key information and be exchanged between different clustering partitions. According to the confusion matrix, COLI chooses the split or re-cluster operation to update the partitioning of each clustering algorithm. Finally, the updated partitions are aggregated by the point majority voting to obtain the final model, which is used to impute missing values.

4.2.1 Exchange of Information

Collaborative clustering is a framework in which different clustering results collaborate to reach a better partitioning of common data. Each partitioning refines itself by exchanging information with other partitionings. Determining the type of information for the exchange within the collaborative process is of paramount importance. The partition structure is one of the most common characters among all clustering methods. Therefore, in order to design a more general algorithm, the proposed method treats the partitions as the key information to exchange.

Let us assume R presents a set of partitioning results, where R_i means the partitioning result obtained by $i-th$ clustering method. c_i^u stands of the $u-th$ cluster in R_i. The confusion matrix is introduced to compare the partitions between two partitioning results. It presents the overlapped samples between two clusters in two different clustering results. The confusion matrix $\Lambda_{i,j}$ between R_i and R_j is defined as follows:

$$\Lambda_{i,j} = \begin{pmatrix} \vartheta_{i,j}^{1,1} & \dots & \vartheta_{i,j}^{1,\eta_j} \\ \vdots & \ddots & \vdots \\ \vartheta_{i,j}^{\eta_i,1} & \dots & \vartheta_{i,j}^{\eta_i,\eta_j} \end{pmatrix} \tag{4.1}$$

where η_i and η_j stand for the number of clusters in R_i and R_j, respectively, and

$\vartheta_{i,j}^{u,t}$ is defined as follows:

$$\vartheta_{i,j}^{u,t} = \frac{\left|c_i^u \cap c_j^t\right|}{\left|c_i^u\right|} \tag{4.2}$$

In this formula, $\left|c_i^u \cap c_j^t\right|$ is the number of samples in u-th cluster of R_i, and t-th cluster of R_j at the same time. The confusion matrix makes it possible to know whether or not the samples of two partitions have been grouped in a similar way or if the two clustering results are dissimilar.

Now the intersection relationship of the clusters in two different clustering results can be find. $o_{i,j}^u$ is used to store a set of clusters in R_j that have overlapped samples with c_i^u. This is the information that will be used to exchange during the collaborative process ($o_{i,j}^u = \{c_j^t, \forall 1 \leq t \leq n_j : \vartheta_{i,j}^{u,t} \neq 0$).

4.2.2 Performance Evaluation

There exist various clustering quality criteria that can be used to evaluate the validity of the partitioning results [46]. However, these quality indexes are independent of the imputation quality. In this situation, the proposed method makes use of an imputation performance index, so-called Normalized Root Mean Square Error (NRMSE), as of partitioning performance and the quality of each cluster in each partitioning. NRMSE is defined as follows:

$$NRMSE = \frac{\|X_{imputed} - X_{original}\|_F}{\|X_{original}\|_F} \tag{4.3}$$

where $X_{imputed}$ is the imputed data, $X_{original}$ is the original complete data, and $\|.\|_F$ stands for the Frobenius norm. Different partitioning results exchange the information and modify their partitions in such a way that they obtain lower NRMSE measures.

4.2.3 Collaborative Process Operators

During the collaborative process, each algorithm modifies its partitioning based on $o_{i,j}^u$. $|o_{i,j}^u|$ presents the number of clusters in $o_{i,j}^u$. If $|o_{i,j}^u| > 1$, it means more than one cluster in R_j have overlapping samples with cluster c_i^u. In this case, the *split* operator is applied to change the partition c_i^u (see Algorithm 4). If $|o_{i,j}^u| = 1$, it means only one cluster of R_j has overlapping samples with the cluster c_i^u. Thus, the *re-cluster* operator is used for c_i^u (see Algorithm 5).

Algorithm 4: The pseudocode of split operation.

```
Input: $c_i^u$ , $R_i$ , $o_{i,j}^u$
Output: New partitioning: $\hat{R}_i$
Definitions:
$o_{i,j}^u[t]$: $t - th$ cluster in $o_{i,j}^u$
begin
    $R_i = R_i \backslash \{c_i^u\}$
    for $t \in [1, |o_{i,j}^u|]$ do
        $R_i = R_i \cup (c_i^u \cap o_{i,j}^u[t])$
    end
    $\hat{R}_i = R_i$
end
```

Once each partitioning has refined to its best that is the highest imputation quality, point majority voting is then used to combine all refined partitioning results into a consensus outcome. This step aims to combine the clustering results that might have different numbers of partitions. For each sample x a voting matrix calculated as:

$$\mathcal{V}(x) = \left\{ \left(v_i^1(x), \ldots, v_i^{\eta_i}(x) \right), 1 \leq i \leq e \right\} \tag{4.4}$$

where η_i is the number of clusters in R_i, e is the number of clustering algorithms and $v_i^1(x)$ is defined as Eq. (8).

$$v_i^u(x) = \sum_{j=1}^{e} vote(x, c_i^u, R_j) \tag{4.5}$$

Algorithm 5: The pseudocode of re-cluster operation.

```
Input: $c_i^u$ , $R_i$
Output: New partitioning: $\hat{R}_i$
Definitions:
$x_t$: Value of t-th sample in $c_i^u$
$\iota_i^\delta$: Centroid of j-th cluster in $R_i$
begin
    $R_i = R_i \backslash \{c_i^u\}$
    for $t \in [1, |c_i^u|]$ do
        /* assign $x_t$ to the closest cluster $c_i^\delta \in R_i$ */ $\lambda = \arg\min_{\delta=1:\eta_i} dist(x_t, \iota_i^\delta)$
        /* update center of $\iota_i^\lambda$ by averaging all of the points that have been
          assigned to it. */ $\iota_i^\lambda = \frac{1}{|\iota_i^\lambda|} \sum_{x \in c_i^\lambda} x$
        $R_i = R_i \cup \{c_i^\lambda\}$
    end
    $\hat{R}_i = R_i$
end
```

where

$$vote(x, c_i^u, R_j) = \begin{cases} 1 \text{ if } \ (i = j \text{ and } x \in c_i^u) \\ \quad \text{or } x \in max(o_{i,j}^u) \\ 0 \text{ else} \end{cases} \tag{4.6}$$

Each sample belongs to the cluster according to the opinion of the majority of different clustering algorithms. So, the clustering result R_i votes for its cluster (c_i^u) and also for the cluster in each other clustering result (R_j) that has the maximum number of overlapped samples with c_i^u. The summation of all values presents the membership degree between x and c_i^u. Finally, the sample x belongs to cluster $\breve{\mathcal{V}}$ which has the maximum membership degree $\breve{\mathcal{V}}(x) = \arg\max_{c_i^u} v_i^u(x)$.

4.2.4 Algorithm and Procedure

step 1: The original incomplete dataset is split into a complete subset and an incomplete subset. The complete subset contains all complete samples, and the incomplete subset contains the samples that have missing values.

step 2: Each clustering algorithm is applied to the complete subset to obtain the partition result independently. The imported clustering algorithm is well-tuned to attain a temporary optimized partitioning, which indicates the best imputation quality.

step 3: Incomplete samples are imputed according to the obtained clustering results. The corresponding values in the nearest cluster centroid are used to impute missing values. The performance of the u-th cluster in R_j (ψ_j^u) is the NRMSE value of imputed samples, which are imputed based on the centroid of c_j^u. The performance of R_j (Ψ_j) is the NRMSE value of all imputed samples in this partition.

step 4: Find the cluster c_r^u, which has the worst imputation performance ($max(\psi_r^u)$). That means this cluster is not suitable to impute missing values. Optimizing the partition of this cluster might get a better imputation quality. Then, another clustering result with the best imputation quality (R_B) is treated as the partition to exchange information with cluster c_r^u.

step 5: The confusion matrix $\Lambda_{r,B}$ and overlapped cluster set $o_{r,B}^u$ are calculated between cluster c_r^u and partition R_B. The number of the overlapped clusters $|o_{r,B}^u|$ are used to decide the operation type for c_r^u. if $|o_{r,B}^u| > 1$, split cluster c_r^u. Otherwise, recluster c_r^u. The updated clustering result is stored in temporary partition $\hat{R}_r$.

step 6: If the imputation performance of temporary partition $\hat{R}_r$ ($\hat{\Psi}_r$) is better than the imputation performance of R_r (Ψ_r), replace R_r with $\hat{R}_r$ and update $\hat{\Psi}_r$ based on the new partition $\hat{R}_r$. Otherwise, keep the original R_r, and modify the performance of cluster c_r^u equal to 0. It means updating the cluster c_r^u cannot achieve a better imputation quality for R_r. Then, the cluster with the second-worst imputation quality is selected to update its partitioning.

step 7: Repeat steps 4 $\sim$ 6 until the performance of clusters (ψ) are all equal to 0. Then, the incomplete subset (X_I) is imputed based on ($\breve{\mathcal{V}}$) which is the final unified partitioning result. The pseudocode of COLI is shown in Algorithm 6.

Algorithm 6: COLI

```
Input: Incomplete dataset: $X$
Output: Completed dataset: $\hat{X}$
Definitions: $X_I$: Incomplete subset.
$X_C$: Complete subset; $R$: Set of clustering results.
$\psi_j^\delta$: Imputation quality of $\delta$-th cluster for j-th partition.
$\Psi_j$: Imputation quality of j-th partition.
$\iota_j^\delta$: Centroid of $\delta$-th cluster in $R_j$
$\eta_j$: Number of clusters in $R_j$
$e$: the number of import cluster algorithms.
begin
    Split $X$ into $X_C$ and $X_I$ subsets.
    $R = \cup_{j=1}^{e} R_j$
    for $j = 1, 2, ..., e$ do
        for $\forall x_i \in X_I$ do
            $\lambda = \arg \min_{\delta=1:\eta_j} dist(x_i, \iota_j^\delta)$
            $\psi_j^\delta = NRMSE(\iota_j^\lambda, x_i)$
        end
        $\Psi_j = \frac{1}{\eta_j} \sum_{\delta=1}^{\eta_j} \psi_j^\delta$
    end
    while $( \exists \psi_j^\delta \neq 0 )$ do
        $[u, r] = \arg \max_{j,\delta} \psi_j^\delta$
        $B = \arg \min_{j=1:e} \Psi_j$
        if $\mid o_{r,B}^u \mid > 1$ then
            $\hat{R}_r = (R_r \backslash \{c_r^u\}) \cup split(c_r^u, R_r, o_{r,B}^u)$
        else
            $\hat{R}_r = (R_r \backslash \{c_r^u\}) \cup recluster(c_r^u, R_r)$
        end
        $R = R \backslash R_r$
        if $\hat{\Psi}_r < \Psi_r$ then
            $R = R \cup \hat{R}_r$ (Update $\Psi_r$)
        else
            $R = R \cup R_r$, $\psi_r^u = 0$
        end
    end
    for $\forall x_i \in X_I$ do
        $\mathcal{U} \leftarrow PointMajorityVote(R)$
        $\hat{x_i} = impute(x_i, \mathcal{U})$
    end
    $\hat{X}_I = \{\hat{x}_1 \cup \hat{x}_2 \cup \hat{x}_3 \cup \ldots \cup \hat{x}_i\}$
    $\hat{X} = \{X_c \cup \hat{X}_I\}$
end
```

4.3 Experimental Results

In this section, the setting of the experiments and the information of experimental datasets are explained in detail. Then, the imputation performance of the proposed method is compared with three clustering-based imputation methods and five other imputation methods in terms of NRMSE.

Table 4.1 – Size of the experimental datasets.

Datasets	# of features	# of samples
4-gauss	12	800
Breast Cancer Wisconsin	9	683
Buddy Move	6	249
BUPA liver disorders	6	345
Dermatology	34	358
Difficult Doughnut	12	400
Divorce	54	170
Glass	9	214
Ionosphere	34	351
Iris	4	150
QSAR aquatic toxicity	9	546
Statlog Heart	13	270
Wholesale customers	8	440
Wine	13	178
Yeast	8	1484

4.3.1 Experimental Settings

The proposed collaborative clustering imputation method (COLI) is compared with three standard clustering-based imputation methods (KMI, FCMI, and PAMI), four classical imputation methods (MI, SkNNI, EMI, and DA) and also the ensemble clustering method (CCI) [44]. These imputation methods have optimized their parameters through pre-experiments. CCI and COLI utilize three individual clustering algorithms, including k-means, fuzzy c-means, and partition around medoids. These clustering algorithms apply the same parameters as KMI, FCMI, and PAMI.

The experiments make use of fifteen publicly available datasets. The number of samples and features are listed in Table 4.1. These datasets did not originally contain missing values. In this case, the INSERT amputation method is applied to create incomplete datasets from each original dataset. These incomplete datasets have three missing mechanisms (MCAR, MAR, and MNAR), four missing ratios (1%, 5%, 10%, and 20%), and five missing distributions (Exponential, Normal, Logistic, Weibull and InverseGaussian). Only MAR and MNAR have different distributions. Thus, each original dataset can generate 44 incomplete datasets (MCAR has four incomplete datasets, MAR and MNAR have 20 incomplete datasets).

4.3.2 Results Analysis

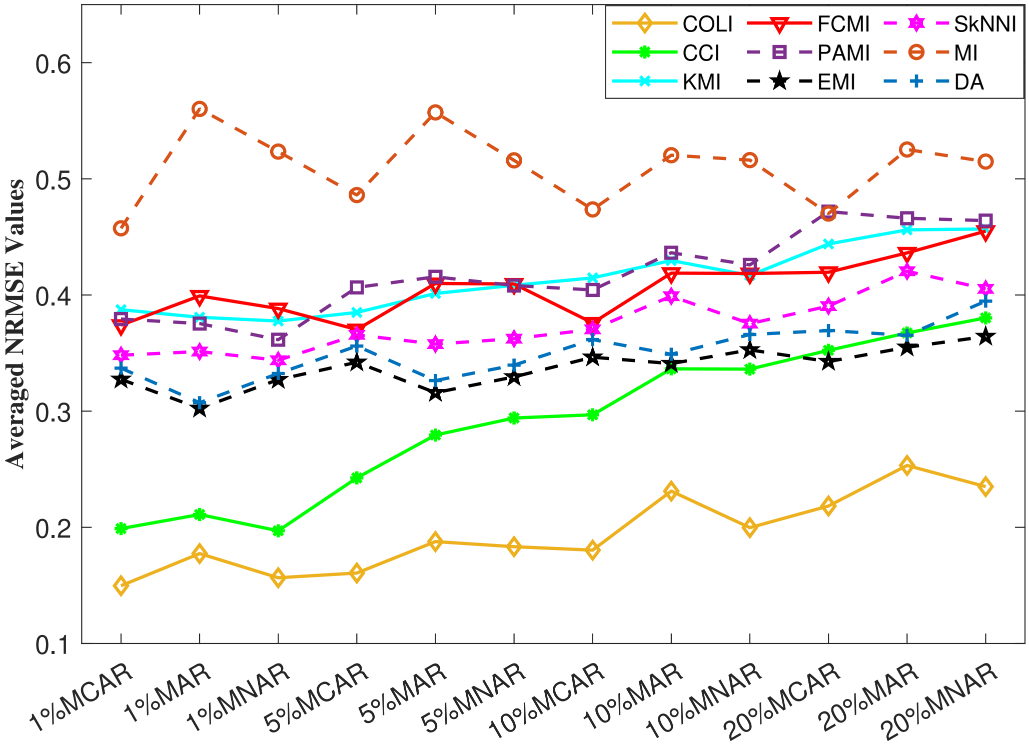


Figure 4.1 – The average NRMSE values obtained by each imputation method over all incomplete datasets for each combination of missing mechanism and missing ratio.

Figure 4.1 shows the average NRMSE values for the combination of missing mechanism and missing ratio. For example, 5%MAR stands for the average NRMSE value for all incomplete datasets with a 5% missing ratio and MAR missing mechanism. In this figure, the proposed method has the lowest averaged NRMSE values overall missing combinations, which demonstrates that the proposed method outperforms its competitors. The proposed method always has a better performance compared with other clustering-based imputation methods. It proves that the proposed collaborative clustering framework is able to improve the quality of the imputation.

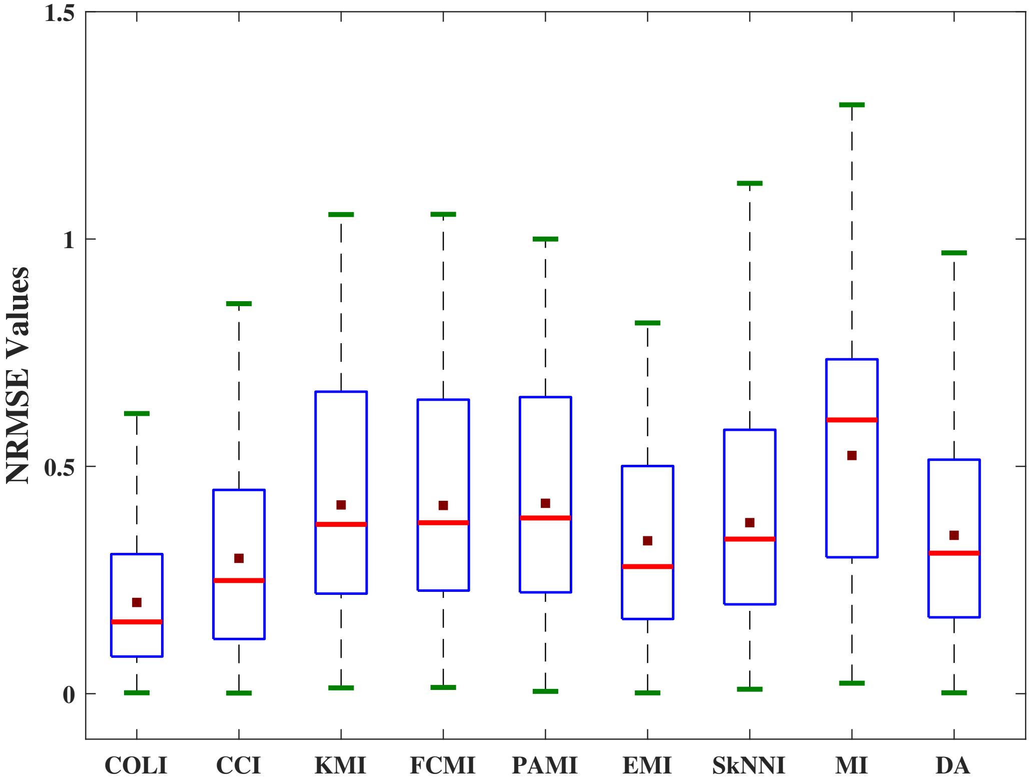


Figure 4.2 – The imputation performance of all incomplete datasets for each imputation method. The red line presents the median value, and the solid square presents the mean value.

Figure 4.2 presents the imputation performance in terms of NRMSE for each imputation method in the form of the boxplot. Obviously, the proposed method has the lowest mean and median values. The size of the box illustrates the variance of imputation performance. The proposed method has the smallest box, which means it has the most stable performance.

Figure 4.3 presents the boxplot of imputation quality in terms of NRMSE obtained by each method overall incomplete datasets with three different missing mechanisms. In this figure, the x-axis presents imputation with different missing mechanisms, the

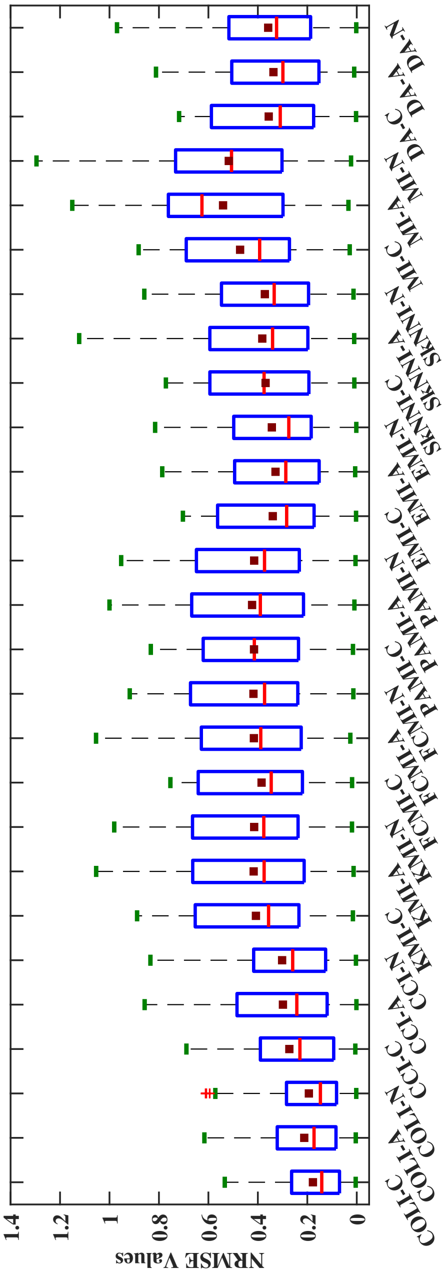


Figure 4.3 – Imputation performance of each method for different missing mechanisms.

name of the imputation method is shown before the dash line, and the letter after the dash line presents the missing mechanism. C stands for MCAR, A stands for MAR, and N stands for MNAR. This figure demonstrates that the proposed method always has the best and the most stable imputation quality. Although the proposed method has worse performance in MAR, comparing to MCAR and MNAR, its performance is still better than any other imputation method. It means that the proposed method is suitable for all missing mechanisms. CCI has the second-best imputation quality. DA and EMI have equal imputation quality with SkNNI for MCAR and MNAR. However, DA and EMI outperforms SkNNI for MAR because these iterative imputation methods are more suitable for missing at random.

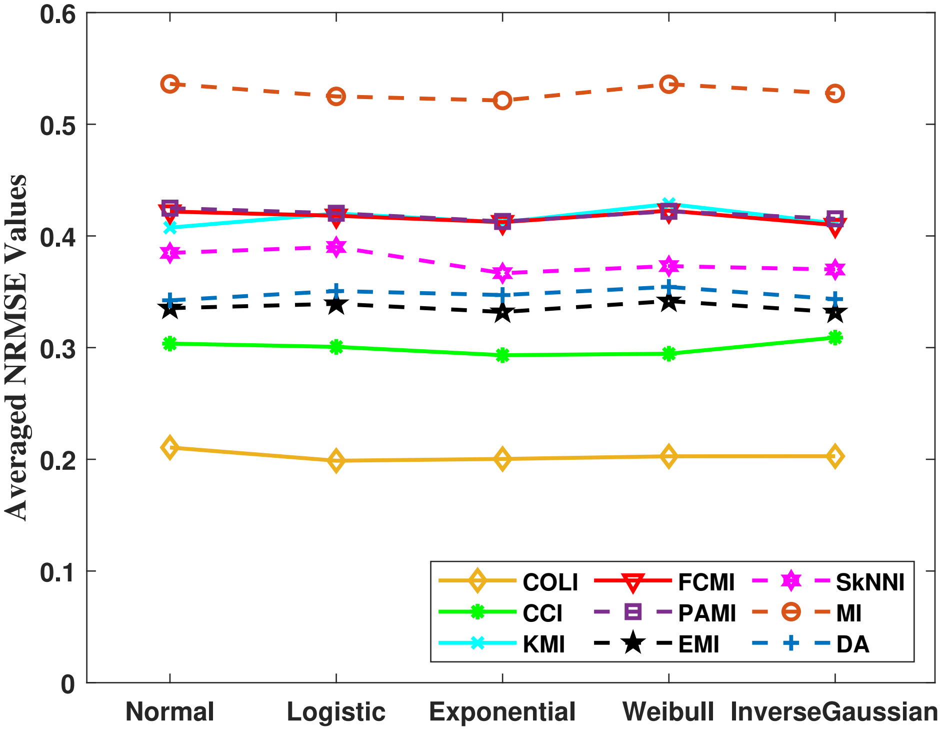


Figure 4.4 – Imputation performance of each method for different missing distributions.

Figure 4.4 presents averaged NRMSE values of different missing distributions for each imputation method. In this figure, the imputation performances of different distributions do not have a significant difference in all imputation methods. That means different probability distribution functions in the amputation process have less influence on the imputation performance. The proposed method still has the best imputation quality comparing to its competitors.

4.4 Summary

This chapter proposes a novel imputation method that is developed from the collaborative clustering framework. The partitioning results obtained by various clustering algorithms communicate with each other to reach a better imputation performance. The imputation quality of the clusters is treated as the information exchanged in the collaborative process. Then, each clustering result updates its partition by two operations, split and recluster. Finally, the updated partitions are aggregated by a consensus function to attain the final partitioning. The proposed method compares with three standard clustering-based imputation methods, four classical imputation methods, and also cooperative clustering imputation. Plenty of incomplete datasets generated from fifteen publicly available datasets are used to evaluate the performance of the proposed method and its competitors. The experimental results demonstrate that the proposed method is an effective method to improve the accuracy of imputation.

Chapter 5

Missing-Tolerant Method

This chapter proposes a novel missing-tolerant method, called MTE-RD. Then, the prediction results of the imputation methods and missing-tolerant methods are analyzed. The significance test and the computational complexity of all missing values treatments are also presented in this chapter. Finally, it gives an answer to the best match for V2X communication datasets with incomplete environments.

5.1 Missing-Tolerant Ensemble

In order to achieve a better prediction result, this thesis proposes a novel missing-tolerant method. It applies Learn^{++} which creates an ensemble of sub-classifiers. Moreover, the predictions of all available sub-classifiers are aggregated by Dempster-Shafer theory that is more effective than the majority vote in information fusion.

5.1.1 Training Phase

MTE-RD creates a set of feature subsets in each iteration to train sub-classifiers. D_t presents the probability distribution for selecting random feature subspace in $t-th$ iteration. D_t can be iteratively updated and it should be normalized at the beginning of each iteration to get the proper distribution. Based on D_t and without replacement, nof features are randomly selected to create $F(t)$ that stores the indices of the selected features. Each sample in the training dataset will be selected for the training subset $\mathcal{X}(t)$ to train $t-th$ sub-classifier C_t, if the feature values of that sample are observed according to $F(t)$. Different sub-classifiers are then trained by the same classification algorithm. If no training sample is complete in $F(t)$, then nof features will be re-selected to create a new $F(t)$. The features that have been selected in the current iteration have less chance to be selected in the next iteration. Hence, the weights of the features in $F(t)$ are reduced by the factor β ($0 < \beta \leq 1$). Consequently, the probability distribution for selecting random feature subspace in the next iteration is updated according to $D_{t+1} = \beta * D_t$. After all sub-classifiers are trained, their effectiveness are presented in a decision profile (θ). The decision profile θ illustrates the support given by the $t-th$ sub-classifier to all classes for a given sample x:

$$\theta(x) = \begin{bmatrix} p_1^1 & p_1^2 & ... & p_1^L \\ p_2^1 & p_2^2 & ... & p_2^L \\ ... & ... & p_t^l & ... \\ p_T^1 & p_T^2 & ... & p_T^L \end{bmatrix} \tag{5.1}$$

where p_t^l presents the probability prediction of sub-classifier t for class l, T stands for the total number of sub-classifiers, and L presents the number of classes.

MTE-RD then calculates the averages of the decision profile for each class by means of the decision template matrix ($\Theta(l)$). The $t-th$ row of the decision template for class l is calculated as $\Theta_t(l) = \frac{1}{N}\sum_{x \in \mathcal{X}^l(t)} \theta_t(x)$. In this equation $\theta_t(x)$ presents $t-th$ row of $\theta(x)$, $\mathcal{X}^l(t)$ represents the samples belonging to class l in subset $\mathcal{X}(t)$. N is the number of samples in subset $\mathcal{X}^l(t)$.

5.1.2 Testing Phase

MTE-RD then finds the available classifiers for the given test sample z based on its no-missing features in the test phase. $F(z)$ is used to store those feature indices that are observed in the target sample z. If all feature indices of $F(t)$ are in the $F(z)$, it means the classifier C_t can be used to predict the test sample z. γ is then created to store the indices of all available sub-classifiers, and τ is the total number of available sub-classifiers. Then, the probability prediction of $t-th$ available classifier for the test sample z is presented by $P_t(z)$. Meanwhile, the rows of all available classifiers $\Theta_\gamma(l)$ in the decision template matrix $\Theta(l)$ form a new decision template matrix $\hat{\Theta}(l)$ for the given test sample z:

$$\hat{\Theta}(l) = \begin{bmatrix} \Theta_{\gamma_1}(l) \\ \Theta_{\gamma_2}(l) \\ \cdots \\ \Theta_{\gamma_\tau}(l) \end{bmatrix} \tag{5.2}$$

$P_t(z)$ is used to present the probability prediction of $t-th$ sub-classifier for the test sample z, $P_t(z) = [p_t^1(z), p_t^2(z), \ldots, p_t^L(z)]$. In order to apply the Dempster-Shafer theory to fuse the evidences, the *proximity* $\Phi_{l,t}(z)$ between class l in the decision template of the $t-th$ sub-classifier $\hat{\Theta}_t(l)$ and the probability prediction $P_t(z)$ is calculated as follows:

$$\Phi_{l,t}(z) = \frac{(1 + \|\hat{\Theta}_t(l) - P_t(z)\|^2)^{-1}}{\sum_{i=1}^{L}(1 + \|\hat{\Theta}_t(i) - P_t(z)\|^2)^{-1}} \tag{5.3}$$

The $belief$ value of sub-classifier P_t can then be computed as follows:

$$b_l(P_t(z)) = \frac{\Phi_{l,t}(z)\prod_{i\neq l}(1 - \Phi_{i,t}(z))}{1 - \Phi_{l,t}(z)[1 - \prod_{i\neq l}(1 - \Phi_{i,t}(z))]} \tag{5.4}$$

Once the $belief$ values for each class are computed, then, the Dempster's rule of combination is used to make the final decision. The final support of each class can be obtained by the product of the $belief$ values from all available sub-classifiers: $\mu_l(z) = K\prod_{t=1}^{\tau} b_l(P_t(z))$, where K is a normalization constant to make the sum of the supports from all sub-classifiers for the $j-th$ class equal to 1. The class that has the highest support is the final prediction for the given test sample z.

The pseudocode of this proposed method is provided in Algorithm 7.

Algorithm 7: MTE-RD.

```
Input: Training data: $X_{train}$ with $n$ features
Testing sample: $z$; Number of subspaces: $T$
Number of features used to train each classifier: $nof$
Output: Final prediction result: $Y$
Definitions:
$L$: Number of classes; $F$: Selected features
$\beta$: Update factor; $\mathcal{X}(t)$: $t-th$ training subset
$\Theta$: Decision template; $\hat{\Theta}$: Decision template for $z$
$P_t$: The probability prediction of the $t$-th classifiers
$\Phi$: Proximity function; $b$: Belief function
$\mu$: Support of each class; $C_t$: $t-th$ sub-classifier
begin
    Initial $D_1(j) = 1/n, \forall j, j = 1, ..., n; \beta = nof/n$
    $t = 1$
    while $t \leq T$ do
        Normalize $D_t$
        /* Select $nof$ features randomly based on $D_t$ */
        $F(t) \leftarrow rand(nof|D_t)$
        if $X_{train}(F(t))$ != $NaN$ then
            $\mathcal{X}(t) = X_{train}(F(t))$
            Train $C_t$ with training subset $\mathcal{X}(t)$
            update $D_{t+1} = \beta * D_t$
            $t = t + 1$
        end
    end
    for $l \in [1, L]$ do
        Calculate $\Theta(l)$
    end
    Find complete features indices $F(z)$ of $z$
    Select applicable $C_t$ and use Eq.2 to create $\hat{\Theta}$
    Get the prediction result $P_t(z)$
    $\Phi(z) \leftarrow proximity(P_t(z), \hat{\Theta})$ (Eq.3)
    $b(P_t(z)) \leftarrow belief(\Phi(z))$ (Eq. 4)
    $\mu(z) \leftarrow combination(b(P_t(z)))$
    Output final prediction $Y$
end
```

5.2 Experimental Results

This section initially compares COLI and CCI with commonly used imputation methods in terms of the imputation performance, i.e., the normalized root mean square error (NRMSE), and, misbehavior detection performance measures, i.e., accuracy and F-measure. It then compares MTE-RD with the state-of-the-art missing-tolerant methods in terms of accuracy and F-measure. All these imputation and tolerant methods are compared together as group or individually to find the best match for misbehavior detection in terms accuracy, F-measure, and computational complexity. These experiments are conducted in a 10-folds cross-validation scheme over incomplete benchmark datasets. The average values of NRMSE, accuracy, and F-measure over ten folds are used in this comparison.

5.2.1 Benchmark Data

Twenty VeReMi datasets with different scenarios are selected for the misbehavior detection. These scenarios include three traffic densities (7 low, 6 moderate, and 7 high densities) and five attacker types. In each dataset, the number of misbehavior messages are far less than the normal messages, which results in class-imbalance datasets, and, thus, F-measure is used as a performance metric for misbehavior detection. To generate incomplete benchmark datasets, original VeReMi datasets are fed to the multi-factor amputation framework (see Section II). INSERT induces missingness over each VeReMi dataset with different missing ratios (1%, 5%, 10%, and 20%), five distributions (Normal, Logistic, Exponential, Weibull, and Inverse Gaussian), and three missing mechanisms (MCAR, MAR, and MNAR), and returns 44 incomplete datasets. Therefore, 880 incomplete datasets are used for the misbehavior detection.

5.2.2 Experimental Result of Imputation Methods

These imputation methods are evaluated by means of the imputation quality (NRMSE) and the prediction performance in terms of accuracy and F-measure. The critical parameters of SkNNI, EMI, DA, and clustering based imputation techniques (KMI, FCMI, and PAMI) are determined within through the cross-validation. The import clustering methods used in the CCI and COLI frameworks are k-means, fuzzy c-means, and partition around medoids. Moreover, according to the previous research [47], the decision tree has the best performance in comparison with other base classifiers, and, thus, the hypothesis set is bounded to decision tree, imputed datasets are fed to the decision tree for the misbehaviour detection.

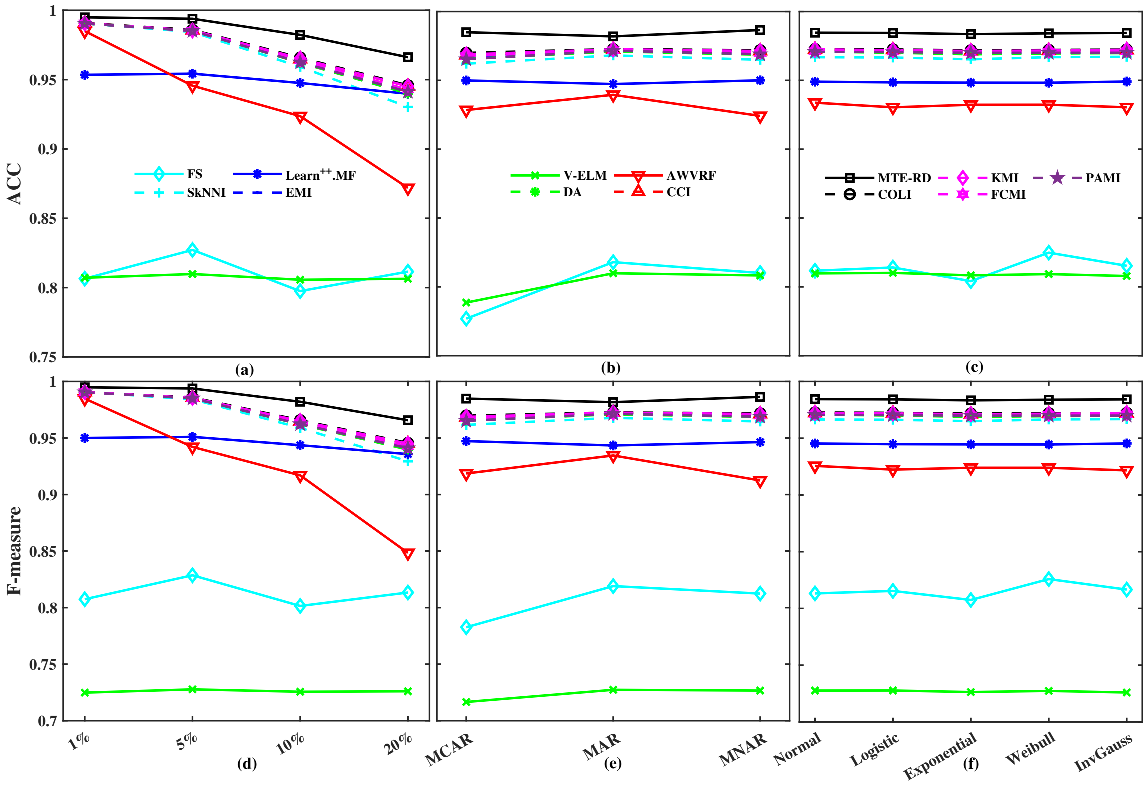


Figure 5.1 – (a) and (d) present the performance measurements of incomplete datasets with different missing ratios. (b) and (e) present the performance measurements of incomplete datasets with different missing mechanisms. (c) and (f) present the performance measurements of incomplete datasets with different missing distributions.

Figure 5.1 shows the detection accuracy and F-measure attained by the decision tree through each imputation method. Each measure, in panels (a-f), is averaged over all benchmark data with different missing ratio, mechanism, and distributions. These imputation methods result in mostly similar performance measures w.r.t. different missing factors. Figure 1 (a) and (d) show that the performance measures decrease by increasing the missing ratio. Figure (b) and (e) show that imputation methods result in slightly better results in the MAR and MNAR missingness compared to MCAR. Figure (c) and (f) show that the detection performance is not very sensitive to the missingness distribution over the benchmark data. In general, Figure 1 (a) to (f) show that imputation methods (dash lines) outperform missing-tolerant methods (solid lines), except MTE-RD, which outperforms all methods in all scenarios.

Table 5.1 reports the averaged measures over all incomplete benchmark data attained through each imputation method. Although COLI and CCI underperform EMI in terms NRMSE, however, they outperform all imputation methods in terms of detection accuracy and F-measure (see bold entries in Table 5.1). In general, clustering-based imputation methods result in better detection performance, but worse imputation quality. SkNNI, EMI, and DA have better imputation quality, but underperform other imputation methods in terms of accuracy and F-measure. Our proposed method, COLI, outperforms all clustering-based imputation methods in terms of NRMSE and all imputation methods in terms of accuracy and F-measure, that are the most important measures for the sake of misbehavior detection.

Table 5.1 – Imputation and classification performance

Name	**Accuracy**	**F-measure**	**NRMSE**
KMI	0.9716	0.9715	0.3336
FCMI	0.9706	0.9705	0.3060
PAMI	0.9698	0.9697	0.3597
SkNNI	0.9661	0.9658	0.1447
EMI	0.9701	0.9699	**0.1401**
DA	0.9694	0.9694	0.1760
CCI	0.9716	0.9715	0.2766
COLI	**0.9721**	**0.9720**	0.2717

5.2.3 Experimental Result of Missing-Tolerant Methods

MTE-RD is compared with four state-of-the-art missing-tolerant methods (FS, Learn^{++}.MF, V-ELM, and AWVRF). FS applies the feature selection method to reduce the dimension of the original dataset. In the experiments, only half of the features are selected by the correlation feature selection (CFS) method. In Learn^{++}.MF, The size of the RSM and the number of features in each feature subsets are decided by pre-experiments. The pre-experiments indicate that the performance does not improve significantly after the size of RSM reached 100, and two features in each feature set have the best performance. Therefore, the critical parameters nof and T are set to 2 and 100. In V-ELM, 11 ELM classifiers are created for each missing pattern, and the activation function is sigmoid. In AWVRF, the random forest has 100 decision trees. The proposed method also applied Learn^{++} to create an ensemble of classifiers. Thus, it has the same parameters as Learn^{++}.MF. Meanwhile, to maintain experimental consistency, the decision tree is treated as the base classifier for each method, except that V-ELM has its own base classifier, ELM.

Figure 5.1 shows the averaged performance measures over all benchmark scenarios with different missing ratios, mechanisms, and distributions attained by each missing-tolerant method (solid lines). MTE-RD outperforms all treatment methods including imputation and tolerant methods over all scenarios with different factors. Although the missing ratio has more influence on the performance of the methods with the random subspace selection mechanism, however, these methods outperform those with the missing pattern mechanism. AWVRF has a better performance with the MAR mechanism, comparing to other two mechanisms. The missing distributions do not affect the performance of the missing-tolerant methods. However, FS has the best (worst) performance over the cases with the Weibull (Exponential) distribution of missingness.

5.2.4 Missing-Tolerant vs Imputation

Figures 5.2 and 5.3 depict the distribution of the attained accuracy and F-measure through each missing data treatment method. Each solid square presents the mean of the performance measure and each red solid line stands for the median value of the performance measures. The green solid line on the top of each box presents the maximum value attained by each method. These missing treatment techniques are ranked w.r.t. the mean values of the attained performance measures as MTE-RD, COLI, KMI, CCI, EMI, FCMI, PAMI, DA, SkNNI, Learn^{++}.MF, AWVRF, FS, and V-ELM, respectively.

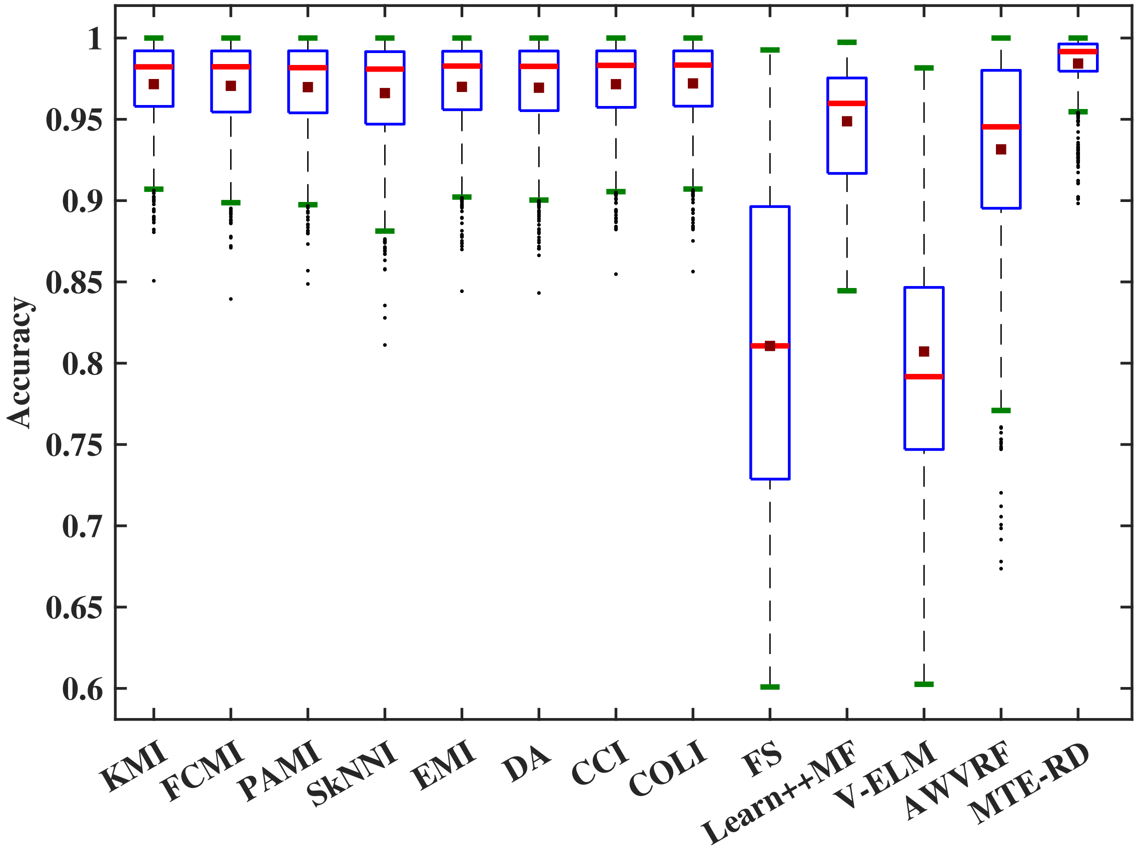


Figure 5.2 – Distribution of the accuracy values obtained by each missing values treatment for all incomplete datasets

The distribution of the attained measures through imputation methods are similar

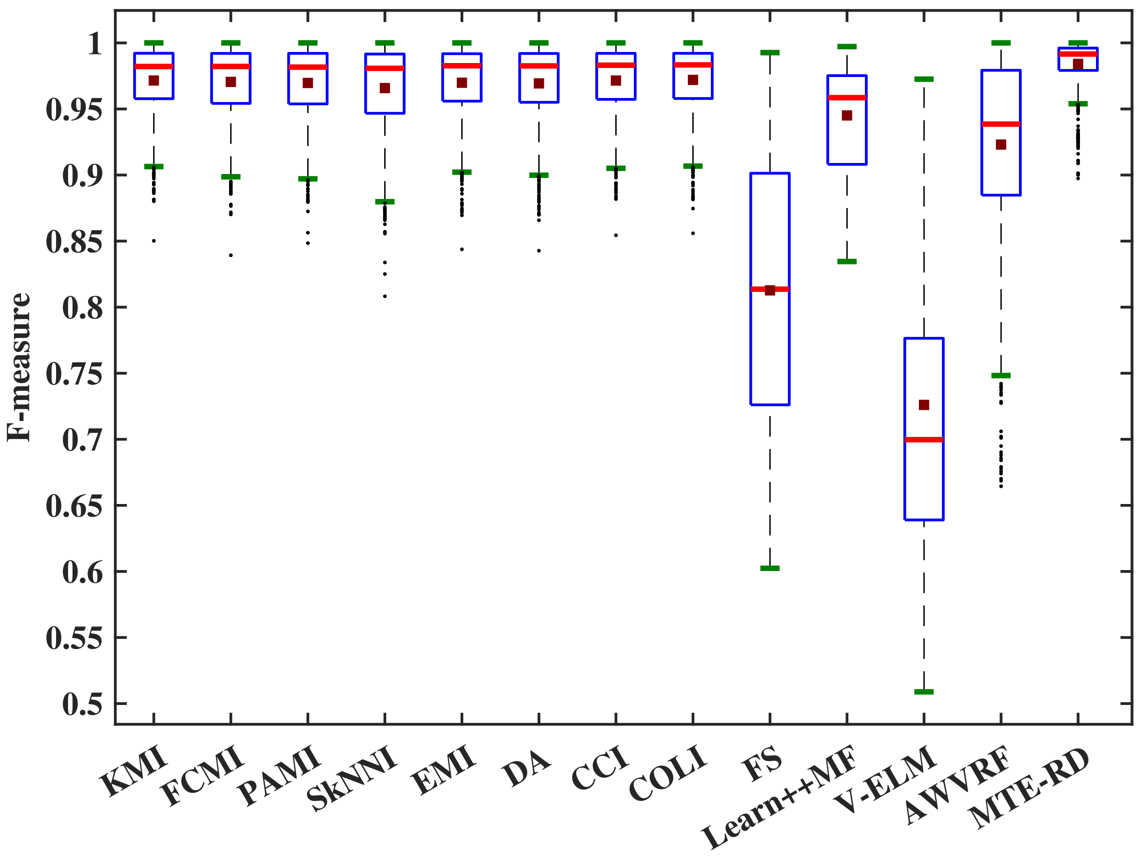


Figure 5.3 – Distribution of the F-measure values obtained by each missing values treatment for all incomplete datasets

to some extent. FS and V-ELM are the most unstable methods and have the lowest performance. That means those tolerant ensemble methods with the missing patterns mechanism are not the prior choice for misbehavior detection over V2X benchmark data. The proposed missing-tolerant method, MTE-RD, has the smallest box and highest mean value, which indicate the maximum stability and highest performance among all competitors. Figures 5.2 and 5.3 also show that imputation methods more stable compared to missing-tolerant methods.

Finally, a two-step test is used to check whether the performance of the proposed methods are significantly different from other methods. The Friedman rank test at the significance level $\alpha = 0.05$ is applied to determine if the performance of one

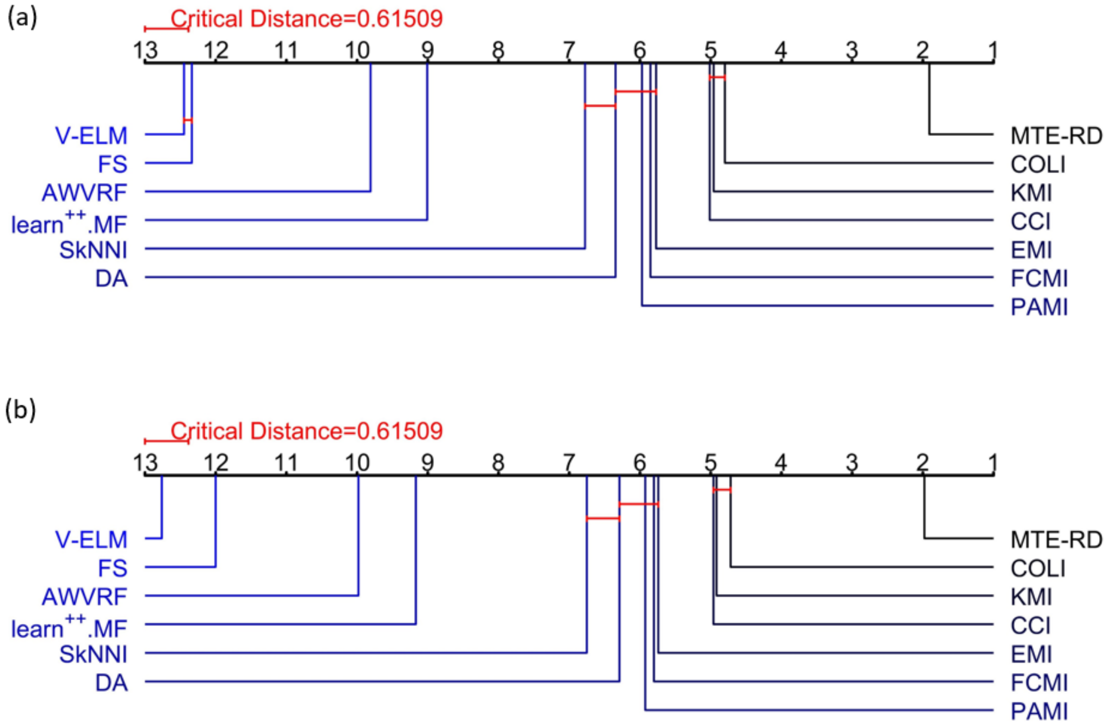


Figure 5.4 – Nemeyi test for (a) accuracy and (b) F-measure

or more methods are significantly different. The attained p-values of the accuracy and F-measure are both less than the significance level of 0.05, which reject the null hypothesis and state that all these methods are not equivalent.

Then, the Nemeyi test with the same significance level is used to compare all methods in a pairwise manner. This test uses a critical difference (CD) diagram to interpret the results, whereas the performance of two methods are significantly different if the gap between the average ranking of these two methods is greater than the critical difference (CD). On the other hand, two methods that do not have a significant difference are connected with the red solid line, where the difference between the average rankings of these two methods is less than the CD value. Figure 5 (a) and (b) depict the CD diagrams in terms of the accuracy and F-measure, respectively. There is no significant difference among the imputation methods. The proposed imputation method COLI has the second high ranks, but it has no significant difference with CCI and KMI. Missing-tolerant methods are significantly different from each other, except V-ELM and FS. MTE-RD has the highest rank and it significantly outperforms all other methods for treating missing values including imputation and tolerant methods.

5.2.5 Complexity Analysis

Table 5.2 reports the computational complexity of all imputation and missing-tolerant methods. The computational complexity of the training and test phases are reported separately. In the complexity formulas, m stands for the number of training samples, n indicates the number of features, η stands for the number of clusters, and d indicates the number of iterations, e indicates the number of import clustering methods to build ensemble, T presents the number of sub-classifiers, g stands for the number of missing patterns, k stands for the number of nearest neighbors, h shows the number of hidden layers, and L is the number of classes.

Table 5.2 – Computational complexity and processing time (ms) for each imputation and missing-tolerant method.

Algorithms	**Training**	**Test**	**Time**	**Rank**
KMI	$O(mn\eta d)$	$O(\eta)$	0.013	1
FCMI	$O(mn\eta d^2)$	$O(\eta)$	0.014	2
PAMI	$O(mn\eta d)$	$O(\eta)$	0.022	3
SkNNI	$O(mnk)$	$O(mk)$	2.10	9
EMI	$O(mn^2 + n^3)$	$O(mn^2 + mn^3)$	0.48	6
DA	$O(mn^2 + mn)$	$O(mn^2 + mn)$	1.90	8
CCI	$O(emn\eta d + m^{2d})$	$O(\eta^e)$	0.023	4
COLI	$O(emn\eta d + m^{ed})$	$O(\eta^e)$	0.025	5
FS	$O(gm)$	$O(g)$	2.20	10
V-ELM	$O(g(mh^2n + h^3))$	$O(g)$	1.30	7
Learn^{++}.MF	$O(Tmn)$	$O(T)$	19.3	12
AWVRF	$O(Tmn)$	$O(T)$	8.30	11
MTE-RD	$O(Tmn)$	$O(L^2T)$	25.4	13

The computational complexity of all methods is reported in Table 5.2. The rightmost column of this table shows the rank of each method w.r.t. the processing time of the test phase. Three clustering-based imputation methods are faster than other methods. CCI and COLI create various partitions and update them to form the ensemble, and, thus, they have higher complexities. Missing-tolerant methods have higher training complexities than imputation methods since the number of subspaces and missing patterns are always larger than the number of clusters. In online detection applications, the computational complexity of the test phase matters. The imputation methods have less process time than the missing-tolerant methods in most cases because the missing-tolerant methods contain many sub-classifiers to make the predictions. However, in the MAR and MNAR missing mechanisms, the number of missing patterns becomes smaller and the processing time of FS and V-ELM becomes shorter. MTE-RD has the highest processing time.

5.3 Summary

This chapter proposes a novel missing-tolerant ensemble (MTE-RD). The generated benchmark data enables a careful comparison between the proposed methods and state-of-the-art imputation and missing-tolerant methods for the sake of misbehavior detection with incomplete V2X benchmark data. This helps not only to find the best method for misbehavior detection but also allows us to compare two major groups of techniques for the treatment of missing data. The attained results indicate that MTE-RD outperforms all other methods in terms of accuracy and F-measure, but has the highest computational complexity. Nevertheless, MTE-RD is the best choice given the fact its prediction time is still shorter than the required time to receive messages. Moreover, the attained results show that, excluding MTE-RD, all imputation methods outperform all missing-tolerant methods in terms of accuracy and F-measure. This shows in general imputation of missing data is better than tolerating them due to the faster processing time as well as having estimations of missing values that can be used for the sake of other applications. Among all these methods, COLI has the second rank and has the closest performance to that of MTE-RD, with less computational complexity.

Chapter 6

Conclusion

The main work of this thesis is to study the strategies for treatments of missing values to find out the best match for misbehavior detection with incomplete V2X communication data. In order to accomplish this task, several imputation and missing-tolerant methods are evaluated by publicly available datasets, VeReMi. In addition, a multi-factor amputation framework is developed to generate incomplete datasets for the purpose of simulating incomplete environments. Then, two novel imputation methods, which are based on cooperative clustering and collaborative clustering, are proposed to achieve better imputation quality. Moreover, a novel missing-tolerant method is also proposed to reach higher accuracy and F-measure. The performance of all missing values treatments is evaluated in terms of accuracy, F-measure, and computational complexity.

In order to comprehensively study the missingness of V2X communication, the introduced INSERT amputation method induces missingness with four missing ratios, three missing mechanisms, and five missing distributions. It can generate 44 incomplete datasets from an original complete dataset in a more scientific and applicable way.

The first proposed missing values treatment is cooperative clustering-based imputation. It generates a set of sub-clusters from different clustering results. The sub-clusters are pairwise merged to reach a better imputation quality. The experiments are conducted in ten publicly available datasets and five V2X communication datasets, and results show that it can significantly improve the imputation quality, comparing with individual clustering-based imputation.

The second proposed imputation method is based on collaborative clustering. It treats the partition of clustering results as the information to be exchanged, and each local clustering result is updated to achieve better imputation quality. This method is compared with cooperative clustering imputation and seven other imputation methods. Fifteen originally complete publicly available datasets are used to test their performance in terms of NRMSE. The results present that the COLI is a much better and more stable imputation method.

The third proposed missing values treatment is a missing-tolerant method. It applies the Learn^{++} to create an ensemble of feature subsets, and the sub-classifier is trained for each feature subset. The testing sample applies all available sub-classifiers to predict its label. The predictions from the different sources are combined with Dempster-Shafer fusion. The method is compared with other four state-of-the-art missing-tolerant methods and eight imputation methods, including CCI and COLI, in twenty selected VeReMi datasets. The experimental results prove that the imputation methods are better than the missing-tolerant methods. However, MTE-RD is significantly better than other missing values treatments in terms of accuracy and F-measure. Moreover, the computational complexity of all missing values treatments is also analyzed. The imputation methods have less processing time, comparing to the missing-tolerant methods. If the application of Misbehavior detection systems have a strict processing time limitation, COLI is the prior choice. Otherwise, MTE-RD methods has the best accuracy for misbehavior detection systems, and it is applicable in most incomplete environments.

References

[1] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, 2020.

[2] YuanYuan Li and L. E. Parker, "Classification with missing data in a wireless sensor network," in *IEEE SoutheastCon 2008*, April 2008, pp. 533–538.

[3] P. Sharma and H. Liu, "A machine learning-based data-centric misbehavior detection model for internet of vehicles," *IEEE Internet Things J.*, 2020.

[4] C. Gautam and V. Ravi, "Data imputation via evolutionary computation, clustering and a neural network," *Neurocomputing*, vol. 156, pp. 134–142, 2015.

[5] R. Razavi-Far, M. Farajzadeh-Zanjani, M. Saif, and S. Chakrabarti, "Correlation clustering imputation for diagnosing attacks and faults with missing power grid data," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1453–1464, 2019.

[6] R. Armina, A. M. Zain, N. A. Ali, and R. Sallehuddin, "A review on missing value estimation using imputation algorithm," in *Journal of Physics: Conference Series*, vol. 892, no. 1, 2017, p. 4.

[7] R. Razavi-Far, S. Chakrabarti, M. Saif, and E. Zio, "An integrated imputation-prediction scheme for prognostics of battery data with missing observations," *Expert Systems with Applications*, vol. 115, pp. 709–723, 2019.

[8] R. Razavi-Far, B. Cheng, M. Saif, and M. Ahmadi, "Similarity-learning information-fusion schemes for missing data imputation," *Knowledge-Based Systems*, vol. 187, p. 104805, 2020.

[9] R. Razavi-Far and M. Saif, “Imputation of missing data for diagnosing sensor faults in a wind turbine,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 99–104.

[10] B. M. Patil, R. C. Joshi, and D. Toshniwal, “Missing value imputation based on k-mean clustering with weighted distance,” in *Int. Conf. on Contemporary Comput.* Springer, 2010, pp. 600–609.

[11] P. D. Allison, *Missing data*, ser. Sage university papers series. Quantitative applications in the social sciences ; no. 07-136. Thousand Oaks, Calif.: Sage Publications, 2001.

[12] Y.-T. Yan, Y.-P. Zhang, J. Chen, and Y.-W. Zhang, “Incomplete data classification with voting based extreme learning machine,” *Neurocomputing*, vol. 193, pp. 167–175, 2016.

[13] R. Polikar, J. DePasquale, H. S. Mohammed, G. Brown, and L. I. Kuncheva, “Learn++.mf: A random subspace approach for the missing feature problem,” *Pattern Recognit*, vol. 43, no. 11, pp. 3817 – 3832, 2010.

[14] J. Xia, S. Zhang, G. Cai, L. Li, Q. Pan, J. Yan, and G. Ning, “Adjusted weight voting algorithm for random forests in handling missing values,” *Pattern Recognit*, vol. 69, pp. 52 – 60, 2017.

[15] W.-C. Lin and C.-F. Tsai, “Missing value imputation: a review and analysis of the literature (2006–2017),” *Artificial Intelligence Review*, vol. 53, p. 1487–1509, 2020.

[16] F. M. Shrive, H. Stuart, H. Quan, and W. A. Ghali, “Dealing with missing data in a multi-question depression scale: a comparison of imputation methods,” *BMC medical research methodology*, vol. 6, no. 1, p. 57, 2006.

[17] U. Garciarena and R. Santana, "An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers," *Expert Systems with Applications*, vol. 89, pp. 52–65, 2017.

[18] A. Folch-Fortuny, F. Arteaga, and A. Ferrer, "Pca model building with missing data: New proposals and a comparative study," *Chemometrics and Intelligent Laboratory Systems*, vol. 146, pp. 77–88, 2015.

[19] R. Razavi-Far and M. Saif, "Imputation of missing data using fuzzy neighborhood density-based clustering," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2016, pp. 1834–1841.

[20] K.-Y. Kim, B.-J. Kim, and G.-S. Yi, "Reuse of imputed data in microarray analysis increases imputation efficiency," *BMC bioinformatics*, vol. 5, no. 1, p. 160, 2004.

[21] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[22] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *Icml*, vol. 1, 2001, pp. 577–584.

[23] S. Zhang, J. Zhang, X. Zhu, Y. Qin, and C. Zhang, "Missing value imputation based on data clustering," in *Transactions on computational science I*. Springer, 2008, pp. 128–138.

[24] Y. UshaRani and P. Sammulal, "A novel approach for imputation of missing attribute values for efficient mining of medical datasets-class based cluster approach," *arXiv preprint arXiv:1605.01010*, 2016.

[25] N. Ankaiah and V. Ravi, "A novel soft computing hybrid for data imputation," in *Proceedings of the International Conference on Data Science (ICDATA)*. Citeseer, 2011, p. 1.

[26] J. Tang, G. Zhang, Y. Wang, H. Wang, and F. Liu, "A hybrid approach to integrate fuzzy c-means based imputation method with genetic algorithm for missing traffic volume data estimation," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 29–40, 2015.

[27] S. Azim and S. Aggarwal, "Hybrid model for data imputation: using fuzzy c means and multi layer perceptron," in *2014 IEEE International Advance Computing Conference (IACC)*. IEEE, 2014, pp. 1281–1285.

[28] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Information Sciences*, vol. 233, pp. 25–35, 2013.

[29] R. Kashef and M. S. Kamel, "Cooperative clustering," *Pattern Recognition*, vol. 43, no. 6, pp. 2315–2329, 2010.

[30] C. T. Tran, M. Zhang, P. Andreae, B. Xue, and L. T. Bui, "An effective and efficient approach to classification with incomplete data," *Knowl Based Syst*, vol. 154, pp. 1 – 16, 2018.

[31] M. Mojarad, H. Parvin, S. Nejatian, and V. Rezaie, "Consensus function based on clusters clustering and iterative fusion of base clusters," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 27, no. 01, pp. 97–120, 2019.

[32] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 1, pp. 160–173, 2007.

[33] S. So, P. Sharma, and J. Petit, “Integrating plausibility checks and machine learning for misbehavior detection in vanet,” in *2018 17th IEEE Int. Conf. on Mach. Learn. and Appl. (ICMLA)*. IEEE, 2018, pp. 564–571.

[34] L. Pan, J. Li *et al.*, “K-nearest neighbor based missing data estimation algorithm in wireless sensor networks,” *Wireless Sensor Network*, vol. 2, no. 02, p. 115, 2010.

[35] R. J. Little and D. B. Rubin, *Statistical analysis with missing data.* John Wiley & Sons, 2019, vol. 793.

[36] M. S. Santos, R. C. Pereira, A. F. Costa, J. P. Soares, J. Santos, and P. H. Abreu, “Generating synthetic missing data: A review by missing mechanism,” *IEEE Access*, vol. 7, pp. 11 651–11 667, 2019.

[37] R. W. van der Heijden, T. Lukaseder, and F. Kargl, “Veremi: A dataset for comparable evaluation of misbehavior detection in vanets,” in *International Conference on Security and Privacy in Communication Systems.* Springer, 2018, pp. 318–337.

[38] A. Cornuéjols, C. Wemmert, P. Gançarski, and Y. Bennani, “Collaborative clustering: Why, when, what and how,” *Inf Fusion*, vol. 39, pp. 81–95, 2018.

[39] G. Forestier, P. Gançarski, and C. Wemmert, “Collaborative clustering with background knowledge,” *Data Knowl Eng*, vol. 69, no. 2, pp. 211–228, 2010.

[40] J.-H. Sublemontier, “Unsupervised collaborative boosting of clustering: An unifying framework for multi-view clustering, multiple consensus clusterings and alternative clustering,” in *The 2013 International Joint Conference on Neural Networks (IJCNN).* IEEE, 2013, pp. 1–8.

[41] J. Sublime, N. Grozavu, Y. Bennani, and A. Cornuéjols, "Collaborative clustering with heterogeneous algorithms," in *2015 International Joint Conference on Neural Networks (IJCNN).* IEEE, 2015, pp. 1–8.

[42] W. Pedrycz, "Collaborative fuzzy clustering," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1675–1686, 2002.

[43] H. Cui, G. Ruan, J. Xue, R. Xie, L. Wang, and X. Feng, "A collaborative divide-and-conquer k-means clustering algorithm for processing large data," in *Proceedings of the 11th ACM Conference on Computing Frontiers*, 2014, pp. 1–10.

[44] D. Wan, R. Razavi-Far, and M. Saif, "Cooperative clustering missing data imputation," in *International Conference on Systems, Man, and Cybernetics (SMC), 2020.* IEEE, 2020.

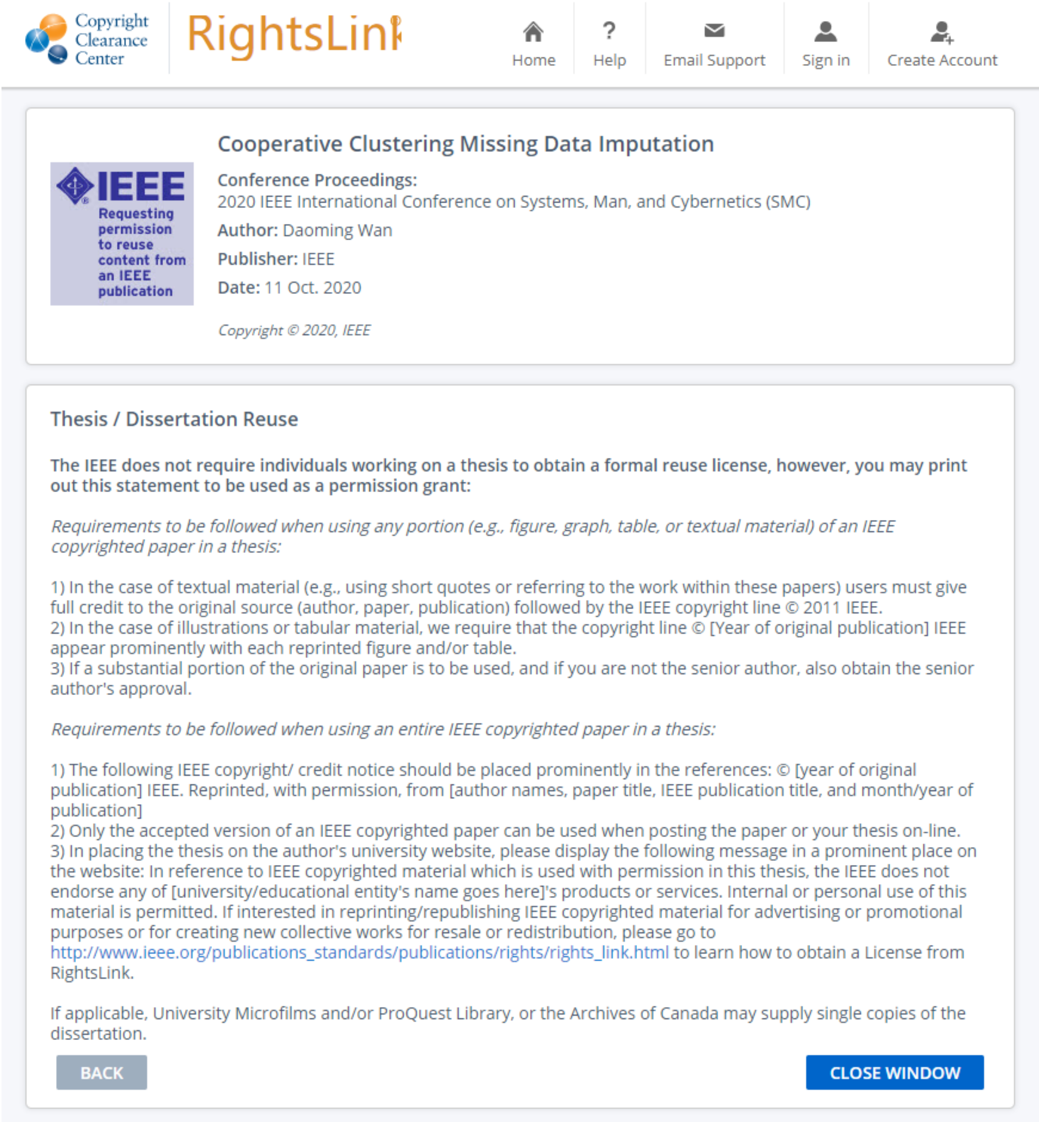
[45] P. Wang and X. Chen, "Three-way ensemble clustering for incomplete data," *IEEE Access*, vol. 8, pp. 91 855–91 864, 2020.

[46] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, no. 15, pp. 3201–3212, 08 2005.

[47] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, "Machine learning approach for multiple misbehavior detection in vanet," in *Int. Conf. on Adv. in Comput. and Commun.* Springer, 2011, pp. 644–653.

Appendix A

Copyright Permissions



Copyright Clearance Center
RightsLink
Home
Help
Email Support
Sign in
Create Account
IEEE
Requesting permission to reuse content from an IEEE publication

Cooperative Clustering Missing Data Imputation

Conference Proceedings:
2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)

Author: Daoming Wan

Publisher: IEEE

Date: 11 Oct. 2020

Copyright © 2020, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Vita Auctoris

Daoming Wan was born in 1993 in Nanchang, China. He persued his undergraduate studies in the Department of Mechanical Engineering at the Southeast University, China, and received B.Sc. degree in mechanical engineering in 2015. He is currently a candidate for the M.A.Sc. degree in Electrical and Computer Engineering at the University of Windsor, Canada and expects to graduate in Spring 2021. He has multidisciplinary background in electrical engineering and mechanical engineering. His research area mainly involves data mining, machine learning, computational intelligence, artificial intelligence, and their application in the V2X communication.