Alleviating Data Sparsity to Enhance AI Models Robustness in IoT Network Security Context

Keshav Sood*, Shigang Liu, Dinh Duc Nha Nguyen, Neeraj Kumar, Bohao Feng, and Shui Yu

***Abstract*—In Internet of Things (IoT) networks, the IoT sensors collect valuable raw data required to sustain Artificial Intelligence (AI) based networks operation. AI models are data driven as they use the data to make accurate network security, management, and operational decisions. Unfortunately, the sensors are deployed in harsh environments which affects the sensor behaviour and eventually the networks' operations. Further, IoT devices are typically vulnerable to a range of malicious events. Therefore, IoT sensor's correct operation including resilience to failure is essential for sustained operations. Naturally, the state variables of time-series data can be changed, i.e., the data streams generated in these situations can be incorrect, incomplete or missing, and sparse presenting a significant challenge for real-time decision-making ability of AI models to make explainable and intelligent management and control decisions. In this paper, we aim to alleviate this fundamental problem to predict the missing and faulty reading correctly so that the decision-making ability of the AI models should not deteriorate in the presence of incorrect, missing, and highly imbalanced data sets. We use a novel approach using fuzzy-based information decomposition to recover the missed data values. We use three data sets, and our preliminary results show that our approach effectively recovers the missed or compromised data samples and help AI models in making accurate decision. Finally, the limitations and future work of this research have been discussed.**

***Index Terms*—Smart networks, Cyber physical systems, Internet of Things, Next generation networks**

I. Introduction

THE use of Artificial Intelligence (AI) is emerging in Internet of Things (IoT) network security and the field is heavily dependent on data science to gather meaningful information to protect data, network users, assets, etc. Data analytics [1], [2] offers a scientific approach to identifying hostile attacks on digital infrastructures to enhance security [3], [4]. For example, to accurately classifying legitimate and non-legitimate network traffic, accurate prediction of anomalies, for monitoring and autonomously identifying security threats, etc., the 'data' is the backbone of AI based network security analytics [5]. Overall we can say that data is an invaluable weapon against intrusions.

However, the *first* challenge is to obtain a complete data for analytics and the *second* challenge is the adaptability of AI models to dynamic IoT sensor network environment [6]. IoTs have given a new attractive surface to attackers and such that any compromised input data, due to both legitimate and cyber-attacks reasons, increases cyber risks which impacts the ability of network security approaches to effectively secure network operations [7], [8]. Incomplete data or if there is considerable missing or compromised data (called *data sparsity* issue), has serious negative impact on the data analyses process in making scientific or intelligent network/application managing and control decisions [9], [10]. To better comprehend this, sparse data is a variable in which the cells do not contain actual data within data analysis. Sparse data is empty or has a zero value. Sparse data is different from missing data because sparse data shows up as empty or zero while missing data does not show what some or any of the values are.[1]

Note that the data sparsity issues or missing data values can occur both due to attack and non-attack events [11]. It is possible that the IoT sensors are faulty and do not generate any data values and therefore the recorded data is not complete or having many cells with zero value or no value. It is also possible that an attacker stealthily compromises IoT sensor readings (by a very small margin), and these manipulated readings bypass the sensor's basic 'faulty data' detection mechanism and propagates to the sensor output undetected [12], [13]. In other words, it means that in real-world the collected data is not always complete, because it is not possible for the sensor to record every sample from every smart product/thing. We encourage readers to read [14] in which the authors argued that sparsity is turning adversarial; they have comprehensively explained different treat models and attack strategies as well.

It is evident that the use of AI will grow in future at a significant rate; and data will remain the key for analytics, [15], [16], [17]. Therefore, there is an urgent need to address the data sparsity issue. Further, it is very challenging to recover the sparsed data in situations of high dimensional data matrix where the percentage of missing or sparse data is significantly high. On top of this the most challenging part is to recover or predict the missed/sparsed values without affecting the AI model's accurate decision-making ability. Other than the AI decision making issue, the missing values in the data set is a critical problem for knowledge discovery, results in large errors and false results [18].

Keshav Sood *(corresponding author) and Dinh Duc Nha Nguyen are with School of Information Technology, Deakin University, Melbourne, Australia. E-mail: keshav.sood@deakin.edu.au, dinh.nguyen@deakin.edu.au

Shigang Liu is with Commonwealth Scientific and Industrial Research Organisation (CSIRO, Data61), Clayton Victoria, Australia. E-mail: shigang.Liu@data61.csiro.au

Neeraj Kumar is with Department of Computer Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India. Email: neeraj.kumar@thapar.edu

Bohao Feng is with Beijing Jiaotong University, China. E-mail: bhfeng@bjtu.edu.cn

Shui Yu is with University of Technology Sydney, Australia. Email: shui.yu@uts.edu.au

[1] https://www.baeldung.com/cs/missing-vs-sparse-data accessed on 06/04/2024

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3525463

To alleviate this issue, the solutions are divided into two categories. 1) In the first category the schemes simply ignore the missing values and conduct the data analysis operation and 2) in the second category, the schemes recover the missed values and then conduct analysis. However, in-spite of having such solutions we observe that simply ignoring the missing values is not always the best approach. Arguably, the predictive models are useful in many prediction applications where the feature selection goes well with a Naïve Bayes approaches. However, it is challenging to build a prediction model when there are missing values (NAN). In counter argument, there is a principled way to deal with missing values [18], for example in Naive Bayes or Gaussian Processes one can choose the best option with the remaining variables and integrate out the missing values. Furthermore, missing values are considered as hidden variables and expectation–maximization (EM) algorithm is used to estimate them. Smola et al. describe a variant of the SVM algorithm which explicitly tackles the problem. To deal with the incomplete data authors have formulated this problem as an optimization problem which extends the Concave Convex Procedure (CCP), and present a simplified and intuitive proof of its convergence [19]. In addition, the randomForestSRC, which implements Breiman's random forests, can effectively tackle missing data, however, speed is a critical problem in many solutions [20].

We propose to apply Fuzzy Information Decomposition theory (FID) method to autonomously compensate for data sparsity in real-time. The two-stage approach predicts the faulty data by FID through weighting and recovery. Weighting produces fuzzy membership functions which can be used to quantify the contribution of the observed data to the missing estimation and recovery allows for estimation of compromised values by considering different contributions of the observed data [21]. Our results show that the proposed scheme is effective in recovering the missed data values and help AI models to maintain its decision-making ability even in the presence of large amount of missed data samples. We use three data sets, Forest Fire, UNSW-15, and PEMS-Bay to evaluate the proposed approach. The approach recovers the missing values based on the contribution of the observed data.

The contributions of this work are as follows.

1) We highlight that the data sparsity (missing or compromised data samples) is a critical issue, causing both due to cyber-attacks and in other legitimate network failure cases, which effects the decision-making process of AI models. We introduce an approach to detect anomalous sensor data while also predicting the missing values. These should help aid AI-based techniques in maintaining their accuracy.
2) We use a Fuzzy Logic concept and investigate a solution to recover/predict the missing values. We use three real world data sets from different domains to evaluate the effectiveness of the solution.
3) Our scheme not only detects the data sparsity but also recovers the missed data values. Using three data sets from three different domains (agriculture, industrial IoT, and transportation) we show that the application of the scheme is wide; not just limited to any single domain.

Benefits:- Overall, our work aims to tackle an important problem, recovering the compromised or missing sensor values in IoT environment by proposing a reasonable theoretical approach (to recovering missing data that will be fed into AI models). Such sensor values can be used for important learning tasks such as activity recognition, app synthesis, and occupancy identification. Therefore, estimating the compromised or missing sensor values can have a practical impact on various areas. We have proposed an approach that can be easily applied in the existing works to mitigate the impact of data sparsity. Theoretically, the results suggest changes to existing AI based IDS design theories and hence will provide a technological advancement in network security field, as from our results we conclude that ignoring data sparsity impacts the decision-making ability of AI models being used in IoT network security domain; in other words, our research suggests a clear relationship between the data driven AI model's performance (accurate decision making ability) and the impact of missed and imbalanced data sets in the input data.

Also, practically it will potentially impact real-world problems by prompting the inclusion of data sparsity's approaches in designing IDS and anomaly detection approaches which have clear practical implications in the network security domain. Overall, the proposed solution is not only necessarily solving a computer security problem, or even an IoT problem, besides the solution appears to be more broadly applicable.

Novelty:- Please note that the goal of the paper is to explore the data sparsity issue in cyber security context which was not well considered by the existing works in their proposed approaches to defend data injection attacks and identifying anomalies as well. Our work is the early ones to exploring and evaluating this critical research problem and this is our major contribution which aspires to make a substantial impact onto the ongoing discourse on enhancing the robustness of the AI models even in the presence of any compromised, sparsed or less amount of data.

The rest of the work is organised as follows: Section II highlights the related works. In Section III, we have discussed the preliminaries of this work. The proposed method and the experimental results are given in Section IV and V, respectively. In Section VI, we have given a comparison of our approach with other works. In Section VII, we discussed the future directions and lessons learned of this research. Finally, Section VIII summarizes the work.

II. Related Work

This section presents a review about the existing works in the 'data sparsity' problem context. We have seen the existing research mainly in intrusion detection systems (IDSs), authentication, and anomaly detection fields. Fundamentally the IDS's are used to identify an illegitimate device (i.e., an intruder) among legitimate devices, the authentication models focus on the identification (recognition) of many legitimate devices (i.e., authentication), finally the anomaly detection systems use machine learning algorithms to identify patterns of behavior that are outside of the norm. In the existing state-of-the-art works (anomaly detection) such as

Peeves [22], HAWatcher [23], HauntedHouse [24], and evasion approach [25], the authors consider IoT anomaly detection from a network perspective or from faulty data injections context. Contrary to the state-of-the-art and existing works, we consider the 'sensor values' instead of other various features (ot limited to timestamps, volume, and size of network packets) with the aim to address the data sparsity issue in cyber security event/context which was previously not considered.

In this IoT data sparsity problem, the proposed approach (FID approach) is fully relevant as it works on the 'data-values' instead of the feature variables used to design anomaly detection approaches [26].

Some pioneer work in this problem domain is given in, [27], [28], and [29]. Very recently, authors in [9] and [30] have highlighted the issue from data injection attacks (DIAs) perspectives. This issue is also cast in a Bayesian framework in which the attack detection is formulated as the likelihood ratio test or alternatively machine learning methods can be employed to learn the geometry of the data generated by the system [31]. Furthermore, in the existing works, researchers used information theory principles to address the data sparsity issue. Primarily, mutual information and Kullback-Leibler (KL) divergence theories are used to characterize the fundamental limits of the attack [32]. Recently authors have proposed novel stealth attack construction scheme with sparsity constraints [30]. They have proposed two heuristic greedy algorithms for the attack construction are proposed. From their numerical analyses they have shown that it is feasible to implement disruptive attacks that have access to small number of data sample observations.

Further in [33], authors proposed a new scheme for false data injection attacks based on a data driven state estimation (SE) model. The scheme is evaluated on a number of standard bus systems. Recently, in cyber physical world, authors in [34] emphasize that false data injection attack (FDIA) has drawn much attention due to its stealthiness. They have taken the advantage of a non-dominated sorting genetic algorithm II (NSGAII) is as the solver of the problem. However it is argued that although the genetic algorithms provide better solution in comparison to the classical models but they do suffer with high computational complexity and decreasing performance w.r.t the increase in the sample or data size [35]. The authors of [34], [36] have not elaborated the work from this perspective, let alone the effectiveness of the proposal with and without the sparsity issue which we have tackled in this work very comprehensively.

Overall, the K-Nearest Neighbours, SOM and MLP are still considered as the state of the art, however the approaches are not suitable for mixed attributes datasets [37]. To justify this, we have provided further discussions. While dealing with missing data problem, the existing work can be divided into two categories: a) imputation or b) data removal.

The first category substitutes reasonable guesses for missing data however valid only when the percentage of missing data is low. The imputation-based methods are dependent on type of missing data for example a) for Missing Completely At Random the Mean, Median, Mode, or any other imputation method is suitable, b) for Missing at Random the Multiple imputation, Regression imputation approaches are suitable, and c) for Missing Not At Random the Pattern Substitution, the Maximum Likelihood estimation imputation approaches are useful [38]. However, we need to run multiple imputation techniques to determine the most robust one eventually this helps to identify any bias and variations from one technique to another. Following this a comparison is needed of the final imputed data to the original non-imputed data to determine the reliability of the imputation approach [39].

The second category is removing the data points, this is straightforward and is only useful when missing values have no importance. In cyber security cases we cannot assume that the missing data is of no importance and hence can be removed. The notion of handling missing data, in the previous works, is that if a data value has a null record, it is deleted. This method typically called complete case analyser or List-wise deletion [40]. Although it is easy to way out the problem however it comes with a heavy price, i.e., by deleting missing sample/point we form a new dataset with reduced sample size causing a higher propensity to worse performance metrics. It is possible that we could simply delete any meaningful information too. For example, in an unfortunate distribution of missing values across data set, the null records in 16% can result in deleting of 80% of dataset [37]. There it does not seem to be an effective approach; we emphasize that the size and the relevance of the attributes are important and should be assessed before attempting deletion.

There are other cons of such approaches are a) information loss which further introduces bias to the final new dataset, b) not appropriate when the data is not missing completely at random, c) not suitable for a large proportion of missing value which impacts the result of all statistical analysis on that data set. Therefore, naturally researchers are advised to collect more data sets and complete data sets when it comes to cyber security for critical decision making, however, this is not always possible. The other methods include K-Nearest Neighbours Imputation based approaches; however, they have poor scalability and not useful in resource-heavy processes. Also, majority of studies do not integrally consider predicting missing value in highly imbalanced data. Further, the existing works mainly focus on either predicting the attack risk based on the missed data values [28], [41] or they construct attack vectors to effectively exploit the correlation between attack variables [30] (see references therein). Further, their objective is to detect the compromised data values. It is worth noting that: a) these solutions are not effective at large scale compromised data and b) after the data is compromised or missed out, they do not recover the missed or compromised data values. Therefore, it is important to timely address the data sparsity issue to strengthen the AI based network security.

We emphasize that our FID approach is more robust handles both the missing data and alleviate the imbalanced data issue, simultaneously, which none of the existing works doing (simultaneously). This is proved by our experiments in which we use different sizes of data sets, the missing values are at different proportions, and we use three ML models. The results verify that the approach is robust enough to handle, large data sets and its performance is likely independent to ML models.

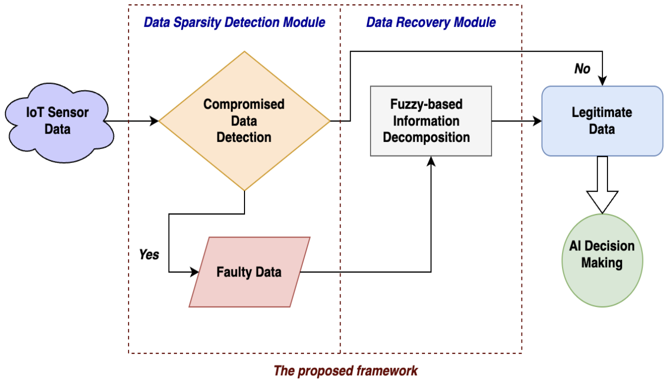


Fig. 1. Framework of the proposed scheme.

III. Preliminaries

In this section, we provide a discussion about the relevance, the importance, the uniqueness, and the novelty of the approach we adopted in this paper, with AI based IoT network security domain. Note that in many scenarios or IoT environments sensors are used to collect readings (we call sensor values), for example many industrial applications in which an array of similar-type sensors is deployed to monitor a specific parameter (e.g., in oil and gas pipelines, smart grid systems, etc.), particularly for example, sensors in smart agriculture sector are used to collect soil temperature, humidity, moisture, etc. Further proximity sensors are typically used to detect the presence or absence of objects near the sensor without any physical contact in order to convert optical images into signals and the typical application areas are found in radar and sonar, biometric devices, medical imaging, etc.

Fundamentally in any IoT network various IoT-enabled sensors are deployed to collect continuous real-time monitoring of sensors' traffic (parameter values, readings or specific parameters). Now, we have taken a reasonable assumption that in the event of a security attack (such as spoofing, man-in-the-middle, or jamming attacks), the adversary will target several sensors. For example, the dataset (sensor values) which is generated by these sensors contains faulty measurements or leads to the faulty data [11]. We emphasize that the faulty data also occurs in case the sensors have a problem by itself due to the impact of the environment or their hardware failure. Therefore, detecting the faulty data is a necessary step to ensure the operation in AI decision making should not alter [42].

To better comprehend this, we assume that an attacker can remotely compromise sensors which also allows them to shut down the sensor for some period. This is very similar case when the sensors are faulty, and they do not generate any value or reading/signal means the AI models at the edge of server side receive little to no data or compromised data for processing which impacts the robustness of the model. The attacker's clear goal is to alter or impact the decision-making ability of the AI models.

Contrary to the state-of-the-art and existing works [9], [43], [44], we consider the 'sensor values' instead of other various features with the aim to address the data sparsity issue in IoT network security context which was previously not considered. The existing works mainly proposed approaches to design IoT node authentication models, defend data injection attacks and identifying anomalies as well. However, our work in cyber security domain is step ahead evaluating this critical research problem which has largely been overlooked; we assume that the sensor values (parameter readings) are not enough, are compromised, or there are many readings with no values or zero values, i.e., data is sparse enough which impacts the decision-making ability (large errors and false results, etc.) of AI models. Hence, the emphasize to apply FID approach is fully relevant as it works on the 'data-values' instead of the feature variables used to design anomaly detection approaches.

IV. Proposed Approach

A novel approach using the Fuzzy-based Information Decomposition (FID) method is investigated to autonomously detect and compensate for data sparsity in real-time. The two-stage approach predicts the faulty data by FID through weighting and recovery. Weighting produces fuzzy membership functions which can be used to quantify the contribution of the collected data to the missing estimation and recovery allows for estimation of compromised values by considering different contributions of the collected data [21]. A high-level framework of the proposed approach is given in Fig. 1. In the proposed framework, the first phase aims to detect the compromised data by using a machine learning model algorithm. Then the second phase employs the FID algorithms to recover the compromised data to ensure the dataset is legitimate for AI decision making operations. The FID in our work firstly detects the data sparsity (see Fig. 1). The FID (having capabilities of sparsity detection and data recovery) is applied in our work in such a way that the proposed approach addresses the compromised data detection, data imbalance and data missing issues simultaneously.

A. The Proposed Framework

The proposed framework has two main modules: Data Sparsity Detection and Data Recovery module. Below, we elaborate each module.

1) Module 1:- Data Sparsity Detection: This module aims to identify compromised data points within the dataset. It works on a combination of historical and real-time data value. The objective here is to detect a subset of compromised data points (outliers) within the dataset. To accomplish the objective, this module further consists of three key stages: preprocessing, model training, and outlier detection. Firstly, the data undergoes the preprocessing stage where it is transformed and scaled. Following which Random Forest (RF), Autoencoder (AE), or K-Nearest Neighbors (KNC or KNN) are trained (using the scaled data) which are employed to predict compromised data points or set of compromised data points as output. In this phase, we also consider the faulty data as outliers and the module first detects it before transferring it to the next stage to "recover" the faulty or missed data. In

Algorithm 1 The work-flow of the proposed framework.

```
Data Sparsity Detection Phase:

Input: $dataset_h$ (historical data for training), $dataset_d$ (historical and real-time data for detection).
Output: A set of compromised data : $d_{sparsity}$
Preprocessing Data:
   - Data transformation:
         $dataset_{numericalH}$ = $dataset_h$.encoder()
         $dataset_{numericalD}$ = $dataset_d$.encoder()
   - Data scaling:
         $dataset_{scaledH}$ = $dataset_{numericalH}$.scale()
         $dataset_{scaledD}$ = $dataset_{numericalD}$.scale()
Detection model: $m_{Outlier-Detection} \in \{RF, AE, KNC\}$
Training stage: $m_{Outlier-Detection}.fit(dataset_{scaledH})$
Detection stage:
result $= m_{Outlier-Detection}.predict(dataset_{scaledD})$
$d_{sparsity}$ = result.detection()
Output: $d_{sparsity}$

Data Recovery Phase:

Input: A set of compromised data : $d_{sparsity}$
Output: Final legitimate dataset : $dataset_{legitimate}$
Determine the number of compromised data:
                              $p = d_{sparsity}.count()$
Identify the min and max values of dataset:
   - Min value: $x = min\ \{\ d_{sparsity}\ \}$
   - Max value: $y = max\ \{\ d_{sparsity}\ \}$
Calculate the step length: $l = (y - x)/p$
Compute intervals: $D_i = [x+(i-1)*l, x+i*l], i \in [1, p-1]$
                   $D_p = [x*(p-1)*l, x+p*l]$
for $i = 1$: p do
      $v_i = (x + (i - 1) * l + x + i * l)/2$
      Weight: if $||q_k - v_i|| > l$ :
                 $w(q_k, v_i) = 0$
            else:
                 $w(q_k, v_i) = 1 - (||q_k - v_i||/l)$
      information decomposition: $n_{ki} = w(q_k, v_i) * q_k$
      Recover compromised data:
            if $\sum_{k \in d_{sparsity}} w(q_k, v_i) = 0$:
                 $recovery_i = d_{sparsity}.mean()$
            else:
                 $recovery_i = \frac{\sum_{k \in d_{sparsity}} n_{ki}}{sum_{k \in d_{sparsity}} w(q_k, v_i)}$
      $dataset_{legitimate}.add(recovery_i)$
Output: $dataset_{legitimate}$
```

our experiments, we have used three machine learning models to determine the outliers in the dataset. After this, the next stage is the recovery of the faulty data using Fuzzy-based Information Decomposition (FID) method. The workflow of this first module is given in the first phase of Algorithm 1. The design, implementation and analysis of this module is given in the following sections.

2) Module 2:- Data Recovery: The second module is the Data Recovery Module which is responsible to recover the compromised data to ensure the dataset is free of compromised or missing data points. We apply the FID algorithm to predict the expected original data points. The overview of the FID algorithm is also shown in Algorithm 1: the number of compromised data points is counted, the minimum and maximum values of the entire dataset are determined, with these values a step length is calculated to partition the data into intervals, within each interval a weight is computed based on a distance measure, this weight guides an information decomposition process aimed at recovering the compromised data. If a data point's weight is zero, it is replaced by the mean of the compromised dataset; otherwise, a weighted sum is calculated to recover the compromised data point. The output of the module is the recovered dataset which is fed to AI models to evaluate the effectiveness of models with an expectation (proved in Section V.B.2) that the recovered dataset will highly maintain the decision-making ability of AI models. Overall, the Data Sparsity Detection Phase aims to identify compromised data points from the received real-time sensor data values, while the subsequent Data Recovery Phase endeavors to recover compromised data using a fuzzy-based information decomposition technique.

B. Fuzzy-based Information Decomposition

The proposed algorithm is explained in this section. Table 1 shows the key mathematical notations used in this work. Assume the column vector a has n features, and p feature values of a are compromised. $a = (a_1, ..., False, ...a_k, ..., False, ..., False, ..., a_n)^T$ where a_k is a data sample value, $False$ presents a compromised data point, and T denotes the transpose of a vector. The V represents the index set of all compromised data values.

$$V = \{k | a_k \neq False, k = 1, 2, ..., n\}. \tag{1}$$

The bounds of the compromised values are defined as lower bound x and the upper bound y. In the paper, x is the minimum value and y is the maximum value of the set of compromised data.

$$\left\{ x = min\{a_k | k \in V\}\ ;\ y = max\{a_k | k \in V\} \right\} \tag{2}$$

We calculate as the equation below to determine an interval D = [x,y]

$$l = (y - x)/p \tag{3}$$

where l is the duration that is computed by p, to be used to find the weight of all checked data. The interval D is divided to p portions for finding the contributions (weights) to predict the original compromised data point values. We have:

$$D_i = [x + (i - 1) * l, x + i * l], i \in [1, p - 1] \tag{4}$$

$$D_p = [x * (p - 1) * l, x + p * l] \tag{5}$$

We note that the intervals are split evenly depending on the total number of compromised data values. In case the length of the p intervals are similar, that is, l. The intervals can be computed as, $D_1 = [x, x+l], D_2 = [x+l, x+2*l], ..., D_{p-1} = [x+(p-2)*l, x+(p-1)*l]$, and $D_p = [x+(p-1)*l, x+p*l]$.

TABLE I
KEY NOTATIONS

Notation	Explanation
$a = (a_1, ..., a_n)$	Row vector
$a = (a_1, ..., a_n)^T$	Transposition of $a = (a_1, ..., a_n)$ column vector
$False$	Compromised data value
p	The number of compromised data
V	Domain of observed feature values
x	The minimum value of the set of compromised data
y	The maximum value of the set of compromised data
D	The interval
l	Length step
p	The feature values
V	Discrete universe of the column vector
v_i	The center of D_i
$w(q_k, v_i)$	The weight
j	The information decomposition

Particularly, $p = 1$ means the number of compromised data is one. Hence, when $p = 1$, there is no need to divide the interval and $D_1 = D = [x, x + l] = [x, y]$.

To compute the contribution weighting, $V = \{v_1, v_2, ..., v_p\}$, we calculate as:

$$v_i = (x + (i - 1) * l + x + i * l)/2, i = 1, 2, ..., p. \quad (6)$$

The discrete universe of a and its value interval have a correlation. So, it can be seen that $v_i - v_{i-1}$ equals to l , and v_i is the centre of D_i.

The aim of the method is recovering the p compromised data points based on the non-compromised data values. There is the correlation of compromised data points and the discrete universe. All the data points contribute to the recovery process to predict the p values. To achieve the goal, we conduct a mapping from a * V to [0, 1]

$$w : a * V \to [0, 1]$$
$$(a_k, v_i) \to w(a_k, v_i)$$

We employ the following equation to compute the contribution weight of each observed data point a_k on D_i:

$$w(a_k, v_i) = \begin{cases} 1 - \frac{||a_k - v_i||}{l}, & \text{if } ||a_k - v_i|| \leq l \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $w(a_k, v_i)$ represents the membership degree of a_k in the D_i in fuzzy set theory. Practically, D_i = { $w(a_k, v_i)|$ k in D }. We emphasize that l provides a range of what kind of compromised data values can contribute to the ith compromised data value. The other ranges such as $||a_k - v_i|| \leq l/2$ might result in dissimilar values. In the proposed method, the general expression is employed, which is $||a_k - v_i|| \leq l$. Then, the contribution weight of an compromised data a_k to D_i which is given as

$$j_{ki} = w(a_k, v_i) * a_k, k \in V \quad (8)$$

In this work, j_{ki} represents as *information decomposition* from a_k to D_i. Now the information decomposition is employed to recover the compromised data.

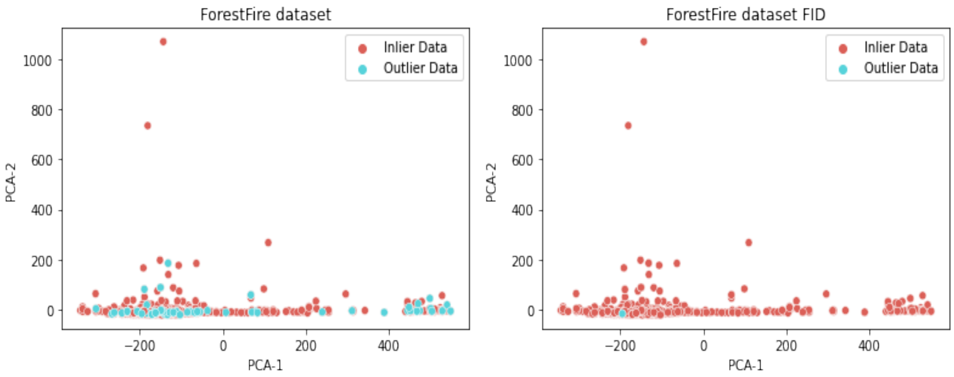
$$j_i = \begin{cases} \overline{a}, & \text{if } \sum_{k \in V} w(q_k, v_i) = 0 \\ \frac{\sum_{k \in V} j_{ki}}{\sum_{k \in V} w(q_k, v_i)}, & \text{otherwise} \end{cases} \quad (9)$$

where $\overline{a}$ is the mean of all data values. $\sum_{k \in V} w(q_k, v_i) = 0$ happens when there is no observed values contribute to D_i. To negotiate with the exception, the compromised data value is defined as the mean of all observed feature values. So, the weighted mean is employed to estimate the compromised data values.

Please note that it is arguable to say why *"if a data point's weight is zero, it is replaced by the mean of the compromised dataset; otherwise, a weighted sum is calculated to recover the compromised data point"*. To counter-argue this, please note that specifically in the missing values recovery process, one scenario is that there may be many missing values. In this case, there might be no values in the interval D_i, which means the data point's weight will be zero, and the missing values will be replaced by the mean of the compromised dataset (also known as the average method for missing values recovery). Otherwise, we will use the proposed equation (7) to calculate the weight for recovering the missing values.

The growing dependence on internet-based services across various aspects of life has made network security a significant concern, as it is crucial for protecting the assets of both sensitive organizations and individuals in the event of an intrusion [45]. Data-driven anomaly detection in network security has long been a hot topic in both industry and academia. Among these techniques, supervised machine learning has been widely accepted and applied for anomaly detection. Supervised machine learning usually trains the prediction model based on the features selected from the original training data, and then the prediction model is used to test any incoming test cases. However, these techniques have not addressed the pending issue of missing data; they simply remove samples with missing values [46]. In the presence of missing data, one needs to improve data quality through data imputation with a separate model. Motivated by this, this work employs the Fuzzy Information Decomposition Scheme to obtain complete data for static-based anomaly detection. The FID approach is based on statistical modelling which is simple mathematical function used to approximate reality and optionally to make predictions from this approximation. Our approach (based on statistical models) is used for finding missing values following which machine learning (ML) models are trained for providing accurate predictions without explicit programming.

The utilization of Fuzzy-Based Information Decomposition presents a novel strategy for addressing missing data concerns within datasets. Unlike conventional statistical methodologies, which typically assume missing data occurs randomly and struggle with complexities in data structures, FID operates on the principles of fuzzy logic to effectively manage uncertainty and imprecision. Through the introduction of soft decision boundaries and adaptability to various missing data mechanisms, FID offers a flexible framework for data imputation. Consequently, FID stands out from conventional imputation techniques by providing a more versatile and resilient approach to handling missing data scenarios. While machine learning and deep learning methods can capture complex patterns in large datasets, they require significant computational resources



(a) (b)

Fig. 2. Data distribution of Forest Fire dataset (a) 10% of anomalies (b) FID recovered the compromised data.

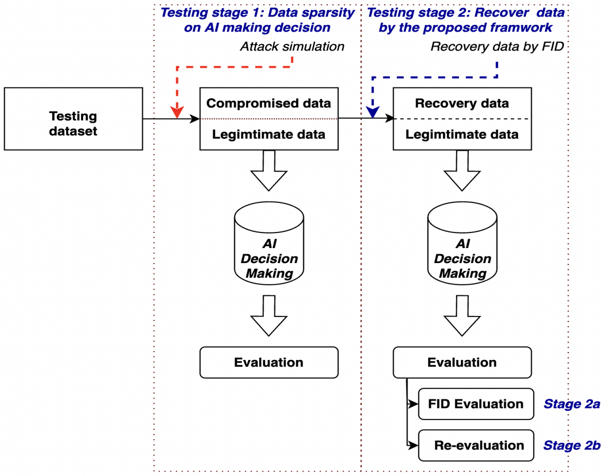


Fig. 3. The workflow of the experiments.

TABLE II
DESCRIPTION OF THREE DATASETS.

Details.	UNSW-NB15	PEMS-Bay	Forest Fire
Number of Instances	175,341	52,116	512
Number of Attributes	44	325	13
Time periods	N/A	January 1–June 30, 2017	January 2000 to December 2003
Compromised attributes	rate & dur	Sensor ID 400001 & 40017	Temperature & Relative Humidity

and often lack interpretability. In contrast, FID provides a more transparent and resource-efficient alternative, though it may not always match the depth of pattern recognition found in these advanced models. Furthermore, compared to multiple imputation and Bayesian methods, which offer robust statistical frameworks but at a high computational cost, FID stands out for its intuitive handling of uncertainty and flexibility, making it a valuable tool in scenarios where data quality and interpretability are crucial.

C. Feasibility Analysis

In this subsection, we want to investigate and show the effectiveness of FID in dealing the missing values. We use Principal component analysis [47] to show the feature space before and after using FID to show FID can recovers the missing values. To visualize the effectiveness of the FID method, the plot of outlier and inlier data in the Forest Fire dataset has been presented in the Fig. 2.

In Fig. 2 (a), there are 10% of anomalies in the dataset. Then, the FID *recovers* the compromised data to original data. Fig. 2 (b) shows how the compromised data is *recovered*. The algorithm for detecting anomalies is KNeighborsClassifier. The Principal Component Analysis (PCA) is employed to reduce the data dimensions to 2-dimensional for plotting purposes. Finally, the information decomposition method ensures that the recovered values can be evenly allocated in the feature space and the proposed method can be used to predict the original value of compromised data.

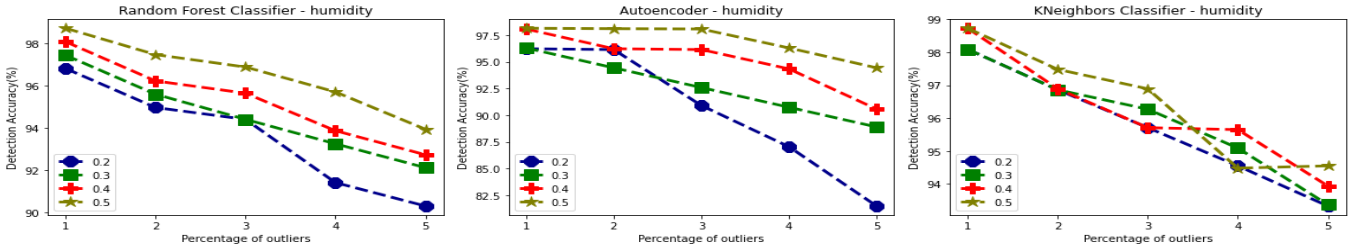
V. PERFORMANCE EVALUATION

We evaluate our approach in two stages: First, we show outlier detection accuracy without any data sparsity. We then include data sparsity, estimate those values with the FID approach, and then show the outlier detection accuracy minimally changes.

A. Experimental Setup

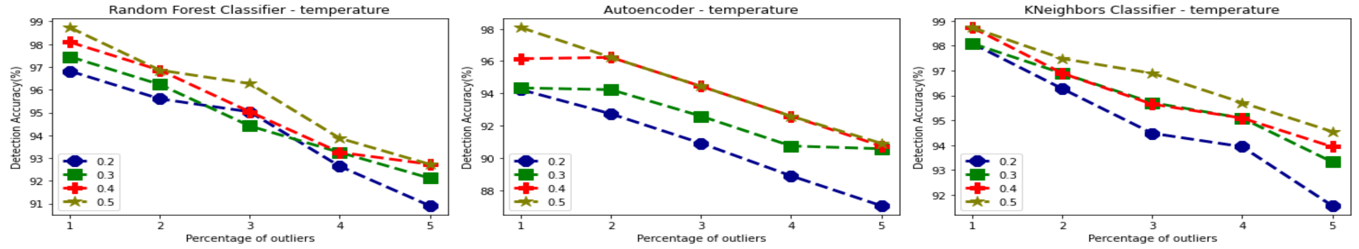
Using open-source tools and data sets we have implemented a Proof-of-concept (PoC) and conducted the performance evaluation of the proposal. The data sets and coding are available via the link provided in the footnote [2]. The experiments are conducted using the GPU on Google Colab is Tesla T4, CUDA version 11.2. A single CPU, A dual cores Intel(R) Xeon(R) CPU @ 2.30GHz on a socket with 2 threads per core, is provided to work with the CUDA cores. The L3 cache reaches 46080K. The mean of CPU frequency is nearly 2300 MHz. The High-RAM mode (26 GB) has been enabled. The testing platform uses Python 3.7.13 and Tensorflow 2.8.0. We have used three data sets. UNSW-NB15 [3], PEMS-Bay[4], and Forest Fire [5]. Table II provides some description of these data sets. The work-flow of our experiments is shown in Fig. 3.

[2] https://github.com/nddducnha/datasparsityFID
[3] https://research.unsw.edu.au/projects/unsw-nb15-dataset
[4] https://dot.ca.gov/programs/traffic-operations/mpr/pems-source
[5] https://archive.ics.uci.edu/ml/datasets/forest+fires



(a) (b) (c)

Fig. 4. Forest Fire dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for humidity outliers.



(a) (b) (c)

Fig. 5. Forest Fire dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for temperature outliers.

To model the security attack scenario, we have randomly selected several IoT neighbours' sensors and then have added Gaussian noise to the data values of those sensors. For this reason, we changed the number of outliers from 1% to 5% of total number of sensors, to observe its impact on the accuracy of the proposed approach. In other words, we changed the area that is targeted by the attacker from 1% to 5% of the whole field network area. Now, to further determine the intensity of the outliers, we have varied the Gaussian noise value. The level of the noise has been changed/varied by considering its standard deviation (sigma) as a percentage of the value of each measurement. Therefore, we have varied this value from 0.2T to 0.5T for compromised features. Naturally as lower the level of noise would be the more difficult it would be to detect relevant outliers. The machine learning model used in the experiments are Random Forest Classifier (RF): $n\_estimators =, random\_state = 0$, Autoencoder (AE): four Neural Network layers, activation function is ”relu”, loss=’msle’, metrics=’mse’, optimizer=’adam’. $EarlyStopping(monitor =' val\_loss', mode =' min')$. $Epochs = 50, batch\_size = 128$, and KNeighbors Classifier (KNC): $n\_neighbors = 5$.

The data is divided into 70% training, 20% testing and 10% for validation. Because the classification problems (outlier detection) do not have a balanced number of examples for each class label. As such, it is desirable to split the dataset into train and test sets in a way that preserves the same proportions of examples in each class as observed in the original dataset. We can achieve this by setting the “stratify” argument to the y component of the original dataset. This will be used by the train_test_split() function to ensure that both the train and test sets have the proportion of examples in each class. The experiment was repeated five times and the results we present in the paper is the average value of these iterations.

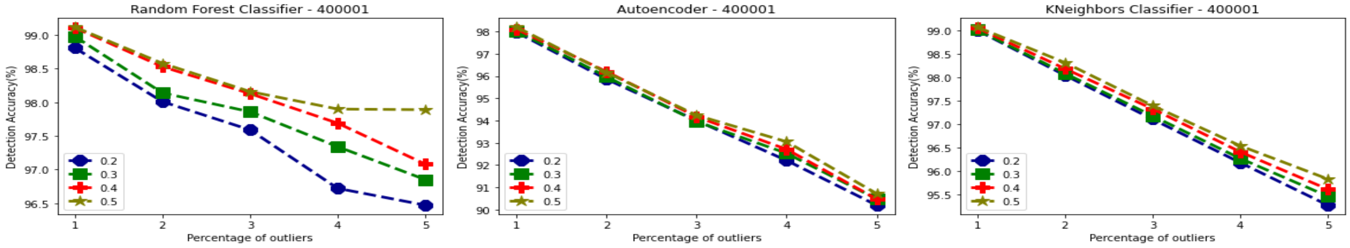
B. Evaluation

In this section, we present the effectiveness of the proposed solution via experiments. We focus on the main scenarios as follow, the high-level view is given in Fig. 3:

- **The impact of data sparsity in AI decision making:** In this scenario (Stage 1), the data sparsity is randomly inserted into three datasets. To present how the data sparsity affects the AI decision making, the detection accuracy is evaluated in several settings.
- **The effectiveness of proposed method to compensate for data sparsity:** (Stage 2a & 2b) We apply the FID methods for data compensation. To prove the effectiveness of the approach, we executed the same setting with the above scenario for detailed comparisons.

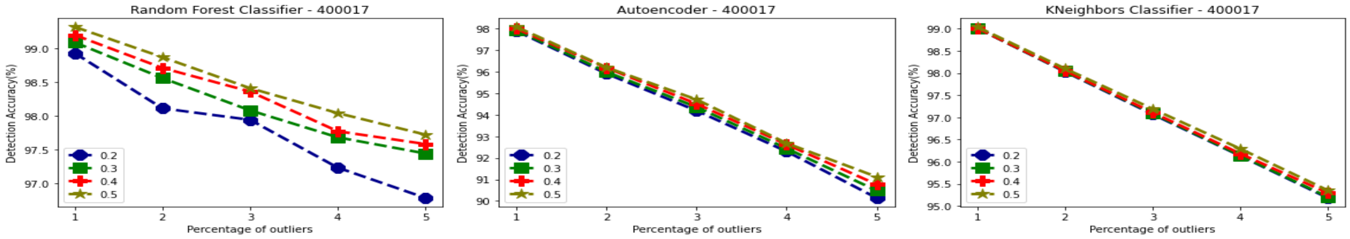
1) The impact of data sparsity in AI decision making: We have shown the recognition accuracy of RF, AE and KNC models on the Forest Fire dataset in Fig. 4 and Fig. 5 for temperature and humidity outliers, respectively. Fig. 6 and Fig. 7 show the outlier detection of three models on the PEMS dataset. In the experiments we choose the two sensors with ID 400001 and 400017 to stimulate the data sparsity. In the UNSW dataset, the features ”rate” and ”dur” are chosen to model outliers. The detection accuracy of three models in the dataset is given in Fig. 8 and Fig. 9.

As we can see, the experimental results showed a similar behaviour for all three algorithms on three datasets. The higher the number of outliers, the lower the outlier accuracy. Moreover, if we keep the fixed number of outliers and increase the sigma, the accuracy is increased. That is the expected result because at low level of noise it is hard to detect the outliers. Furthermore, the outlier accuracy of UNSW and PEMS dataset



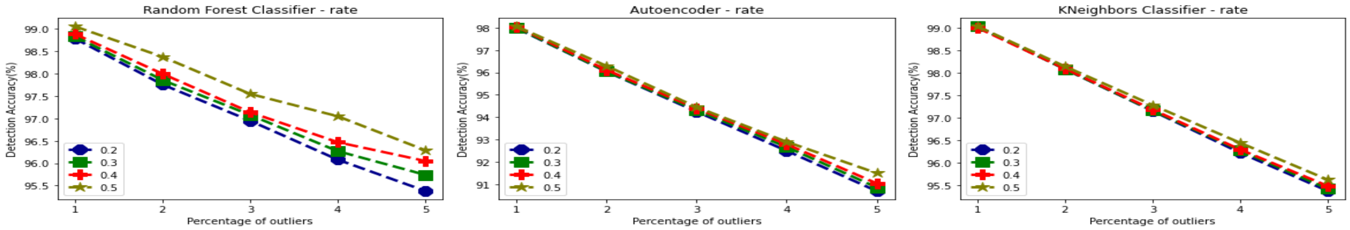
(a) (b) (c)

Fig. 6. PEMS dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for sensor ID 400001 outliers.



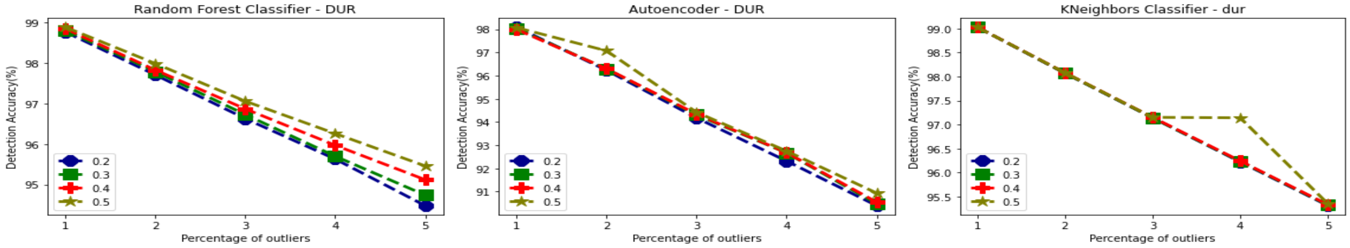
(a) (b) (c)

Fig. 7. PEMS dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for sensor ID 400017 outliers.



(a) (b) (c)

Fig. 8. UNSW dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for ”rate” outliers.



(a) (b) (c)

Fig. 9. UNSW dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for ”dur” outliers.

are higher than the Forest Fire dataset, this is due to the size of the dataset. The size of Forest Fire is very small, so it affects the operations of machine learning models. The impact of data sparsity on various sizes of datasets can be seen.

In terms of machine learning algorithms, Random Forest Classifiers performs better than Autoencoder and KNeighbors Classifier. The RF model achieves highest accuracy detection in most scenarios. On the larger dataset (UNSW and PEMS dataset), the RF model also remains stable when we increase the outliers which are in the range of 96.29% to 99.06% and 97.89% to 99.12%. On the small dataset (Forest Fire dataset), KNC is the most stable in the range of 99.45% to 98.73%. Autoencoder have poor stability when the outlier increases. The detection accuracy jumps down rapidly when the number of outliers increased. Therefore, the AE model is less sensitive to the intensity of outliers than RF and KNC. The other classification metrics of RF on Forest Fire dataset (Table III), Autoencoder on PEMS dataset (Table IV) and KNeighbors Classifier on UNSW dataset (Table V). In summary, the data sparsity does affect AI decision making.

TABLE III
CLASSIFICATION METRIC OF RANDOM FOREST CLASSIFIER IN VARIOUS PERCENTAGE OF OUTLIERS OF FOREST FIRE DATASET.

Metrics\Outliers	**1%**	**2%**	**3%**	**4%**	**5%**	**1%**	**2%**	**3%**	**4%**	**5%**
Compromised features	Temperature and $\sigma = 0.5$					Relative Humidity and $\sigma = 0.5$				
Accuracy(%)	98.73	96.86	96.27	93.87	92.73	98.73	97.48	96.89	95.71	93.94
F1-score(%)	99.36	98.40	98.10	96.83	96.23	99.36	98.73	98.42	97.81	96.88
Precision(%)	98.73	97.47	96.88	95.63	94.44	98.73	97.48	96.89	95.71	94.51
Recall (Sensitivity)(%)	99.35	99.35	99.36	98.08	98.08	99.35	98.71	98.72	98.08	98.08
Compromised features	Temperature and $\sigma = 0.2$					Relative Humidity and $\sigma = 0.2$				
Accuracy(%)	96.82	95.6	95.03	92.64	90.91	96.82	94.97	94.41	91.41	90.3
F1-score(%)	98.39	97.75	97.45	96.18	95.24	98.38	97.42	97.12	95.51	94.90
Precision(%)	98.70	97.44	96.83	95.57	94.34	98.70	97.42	96.81	95.51	94.30
Recall (Sensitivity)(%)	98.06	98.06	98.07	96.79	96.16	98.06	97.41	97.44	95.51	95.51

TABLE IV
CLASSIFICATION METRIC OF AUTOENCODER IN VARIOUS PERCENTAGE OF OUTLIERS OF PEMS DATASET.

Metrics\Outliers	**1%**	**2%**	**3%**	**4%**	**5%**	**1%**	**2%**	**3%**	**4%**	**5%**
Compromised features	Sensor ID 400001 and $\sigma = 0.5$					Sensor ID 400017 and $\sigma = 0.5$				
Accuracy(%)	98.2	96.16	94.24	93.05	90.72	98.08	96.2	94.71	92.68	91.12
F1-score(%)	99.09	98.04	97.03	96.39	95.11	99.03	98.06	97.28	96.19	95.34
Precision(%)	99.01	98.08	97.13	96.25	95.26	98.99	98.06	97.10	96.13	95.31
Recall (Sensitivity)(%)	99.17	98.00	96.94	96.53	94.98	99.07	98.06	97.46	96.26	95.37
Compromised features	Sensor ID 400001 and $\sigma = 0.2$					Sensor ID 400017 and $\sigma = 0.2$				
Accuracy(%)	97.97	95.86	94	92.21	90.2	97.89	95.91	94.19	92.3	90.14
F1-score(%)	98.97	97.88	96.90	95.94	94.84	98.93	97.91	97.04	95.99	94.79
Precision(%)	98.99	98.02	97.03	96.13	95.16	98.99	98.02	97.07	96.19	95.32
Recall (Sensitivity)(%)	98.95	97.75	96.78	95.76	94.52	98.88	97.81	96.94	95.79	94.28

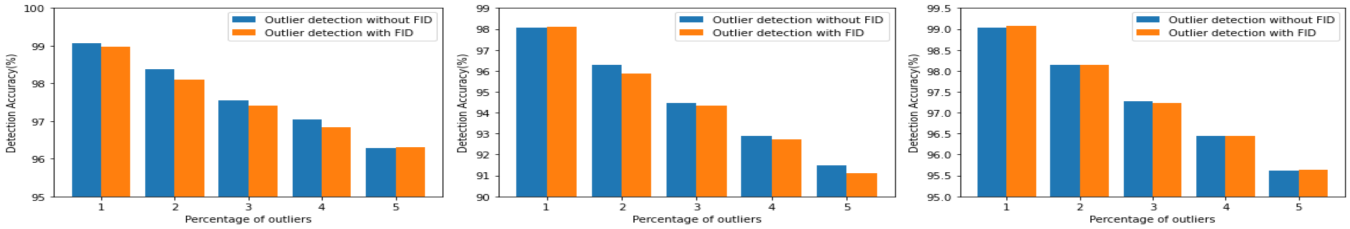
TABLE V
CLASSIFICATION METRIC OF KNEIGHBORS CLASSIFIER IN VARIOUS PERCENTAGE OF OUTLIERS OF UNSW DATASET.

Metrics\Outliers	**1%**	**2%**	**3%**	**4%**	**5%**	**1%**	**2%**	**3%**	**4%**	**5%**
Compromised features	”rate” feature and $\sigma = 0.5$					”dur” feature and $\sigma = 0.5$				
Accuracy(%)	99.03	98.14	97.28	96.44	95.62	99.03	98.08	97.15	97.14	95.35
F1-score(%)	99.51	99.05	98.62	98.18	97.75	99.51	99.03	98.55	98.55	97.62
Precision(%)	99.03	98.14	97.29	96.45	95.62	99.03	98.08	97.15	97.15	95.37
Recall (Sensitivity)(%)	99.99	99.99	99.98	99.98	99.97	1	99.99	99.99	99.99	99.96
Compromised features	”rate” feature and $\sigma = 0.2$					”dur” feature and $\sigma = 0.2$				
Accuracy(%)	99.03	98.08	97.15	96.22	95.37	99.03	98.07	97.14	96.22	95.31
F1-score(%)	99.51	99.03	98.55	98.07	97.63	99.51	99.03	98.55	98.07	97.59
Precision(%)	99.03	98.08	97.16	96.24	95.39	99.04	98.076	97.14	96.23	95.36
Recall (Sensitivity)(%)	1	99.99	99.99	99.97	99.97	1	99.99	99.99	99.98	99.95

2) The effectiveness of proposed method to compensate for data sparsity: We apply our method to compensate for the data sparsity. To demonstrate how our method can mitigate the data sparsity, we firstly apply FID on the dataset with the outlier as the above scenarios. After that, the experiments with the same settings with above scenarios are conducted. We then compared case by case to see how similar the dataset was with/without applying FID. For example, we assume that the *original outlier dataset* has 1% of outlier (as the previous experiments). We increase the outlier to 6% to simulate the data sparsity, it means that we have 5% of compromised data samples in the example. Then we apply the FID method to predict the 5% compromised data samples. After this step, we have a *new outlier dataset* which also contains 1% outliers as the *original outlier dataset*. Then, we evaluate the *original outlier dataset* and *new outlier dataset* in AI decision making operations. If the result is similar or nearly the same, we can conclude that the proposed method is successful to “fix” the compromised data in AI decision making.

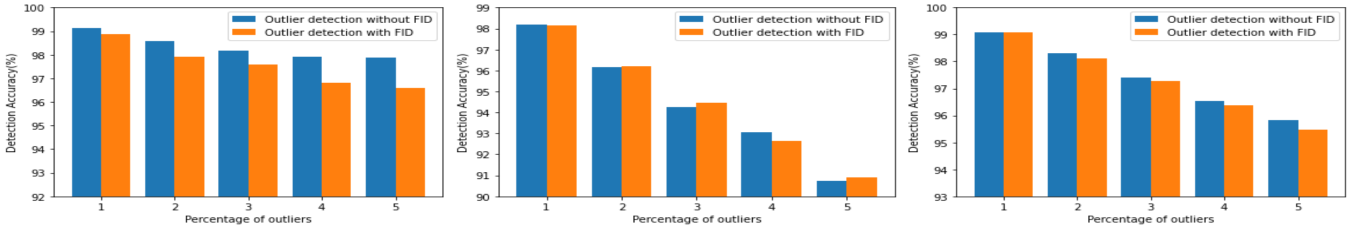
We evaluate our method with data compensation with various scenarios. We use same three machine learning models (RF, AE, KNC). The experiments are conducted on three dataset and the percentage of outliers are chosen from 1% to 5% (with the step of 1%), the level of noise value varies from 0.2T to 0.5T. The same compromised features with previous experiments of each dataset are considered. Firstly, we keep the sigma at 0.5T and maintain the outlier percentages from the range of 1% to 5%. Then, we test the three models with the *new outlier dataset* and compare the result with the *original outlier dataset*. The comparisons are given in Fig. 10 for UNSW dataset, Fig. 11 for PEMS dataset and Fig. 12 for Forest Fire dataset. Note that only one feature of these datasets is used to get the results this is due to the space limitation. We note that the difference between the *original outlier dataset* and the *new outlier dataset* is not significant. KNC gives more stable recognition accuracy than other machine learning models on three datasets. On the large dataset (UNSW and PEMS), our method has performed better to compensate for compromised data than over small datasets. It can be explained that the more data samples support the more accuracy to predict the compromised data.

To prove that our method alleviates the data sparsity issue in AI decision making, we conducted further experiments with all the same scenarios as taken in previous experiments. The results are shown from Fig. 13 to Fig. 18. Please see the results in the Supplementary files. Comparing to the Fig. 4 to Fig. 9, we note the same behaviour between the *original*



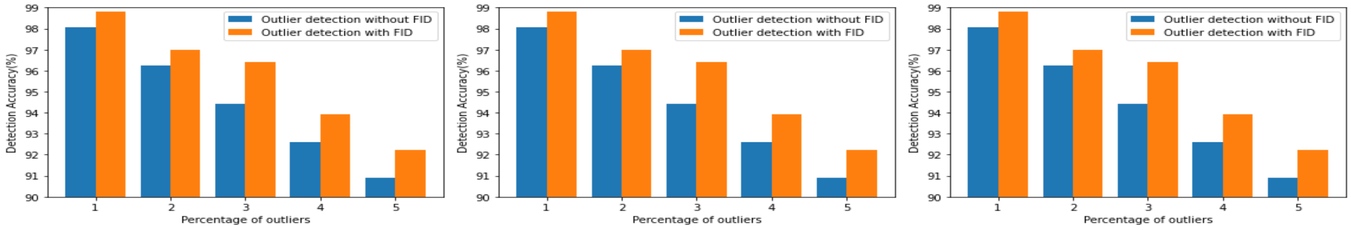
(a) (b) (c)

Fig. 10. UNSW dataset: comparison of recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for ”rate” outliers with and without FID method at sigma = 0.5.



(a) (b) (c)

Fig. 11. PEMS dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for sensor ID 400001 outliers with and without FID method at sigma = 0.5.



(a) (b) (c)

Fig. 12. ForestFire dataset: recognition accuracy of (a) Random Forest Classifier, (b) Autoencoder, and (c) KNeighbors Classifier for temperature outliers with and without FID method at sigma = 0.5.

outlier dataset and *new outlier dataset*. The trends of the graphs are the same. In other words, the proposed method can compensate for the data sparsity successfully.

VI. Comparison with the State-of-the-Art

In this section, we compare our approach with three advanced data recovery methods designed to handle missing and uneven data in different ways. The first method, described in [48], presents a new algorithm called the Flexible EM (FEM), which is specifically designed to deal with missing data; non-Gaussian data compared to traditional models. Standard EM algorithms for Gaussian Mixture Models (GMM) often struggle when the data is inconsistent or contains outliers. To address the issue, the FEM algorithm uses a special type of Gaussian distribution, making it more resilient to outliers while still effectively handling missing data. The FEM algorithm works by repeatedly refining its estimates of data patterns through two main steps: Expectation (E) and Maximization (M). It begins by setting initial guesses for the data's structure using a technique called K-means clustering. In the E-step, it calculates how likely each data point is to belong to each group, using a robust statistical approach that helps handle outliers. The M-step then updates the initial guesses to better fit the observed data by recalculating the group weights, averages, and spread based on the updated likelihoods. This process continues until the algorithm finds the best fit, resulting in a reliable model that can effectively fill in missing data and manage noise. In our experiments, we use the FEM algorithm to recover data points affected by outliers.

The second solution, described in [49], is called Switching Triple-Weight SMOTE (NSS) and uses a multi-step process to recover missing data. Firstly, to reproduce this work at our side, we use Lagrange Non-Negative Matrix Factorization (LNMF) to fill in missing values, preserving the original data's structure by breaking down observed data into hidden features. This step ensures that missing values are reconstructed accurately while maintaining non-negativity, crucial for datasets with inherent non-negative features. After imputing the missing data, we map the dataset into an Empirical Feature Space (EFS), which helps separate the data more

TABLE VI
COMPARISON OF DIFFERENT STATE-OF-THE-ART SOLUTIONS (FEM [48], SMOTE [49], AND AMSA-VAE [50]) WITH THE PROPOSED APPROACH.

Dataset	Metric	FEM	SMOTE	AMSA-VAE	Proposed Solution
Forest Fire	Accuracy	90.1	91.6	92.9	93.85
	F1	92.0	93.5	95.8	96.62
	Precision	89.5	91.2	93.3	94.42
	Recall	90.5	93.4	94.6	97.97
UNSW	Accuracy	90.2	91.3	92.4	93.85
	F1	92.2	93.3	95.7	96.62
	Precision	88.9	90.5	92.8	94.42
	Recall	92.3	93.2	94.3	97.97
PEMS	Accuracy	88.5	89.5	90.8	91.06
	F1	90.5	91.2	92.4	95.21
	Precision	91.1	92.9	93.9	95.18
	Recall	90.3	91.2	93.1	94.12

effectively and reduces unnecessary features, making the data cleaner and easier to work with. Next, we apply fuzzy c-means clustering to divide the data into groups based on their characteristics. The clusters are then analyzed using a triple-weight strategy that considers three key factors: the distance between clusters, the size of each cluster, and how spread out the data points are within each cluster. This strategy is used to determine the number of synthetic samples to generate for each cluster. Depending on the cluster's distribution type (Gaussian or Uniform), the approach adapt the oversampling method to ensure that the new synthetic data matches the natural variability and characteristics of the original data.

Lastly, the Adaptive Multi-Head Self-Attention VAE (AMSA-VAE) [50] leverages an Adaptive Multi-Head Self-Attention Variational Autoencoder (AMSA-VAE) to dynamically fill missing data. This approach integrates adaptive multi-head self-attention (AMSA) layers within the encoder and decoder of a Variational Autoencoder (VAE) framework, allowing the model to extract relevant features from input data sequences and reconstruct missing values. The AMSA layer enhances the VAE by using multiple attention heads to focus on different aspects of the input, capturing complex dependencies and relationships that standard models might miss. During encoding, AMSA extracts key patterns from the input data, while the reparameterization trick in the latent space allows the model to generate realistic estimates of missing values. The decoder, equipped with AMSA, then reconstructs the input data, ensuring the imputed values align closely with the original data distribution. The combination of adaptive attention mechanisms and probabilistic modeling enables the AMSA-VAE to handle noisy, non-Gaussian data effectively, making it a robust solution for missing data imputation in dynamic and complex datasets. We leverage the AMSA-VAE model as follows: first, we pre-process the data to standardize it using a scaler approach, ensuring that the data is normalized for better model performance. The AMSA-VAE model is then constructed with adaptive multi-head self-attention (AMSA) layers integrated into both the encoder and decoder sections. The encoder applies AMSA to dynamically capture patterns and dependencies from the input sequences, while the latent space is sampled using the reparameterization trick to generate realistic estimates for missing data points. The decoder, also equipped with AMSA layers, reconstructs the input by aligning the filled data with the original data distribution, thereby effectively restoring missing values. After training the model on the scaled data, it is used to predict and impute missing entries, updating the original dataset with the reconstructed values.

We implemented the three solutions on the same Google Colab platform, as described, and tested them under scenarios where 6% of the data samples were designated as outliers. The solutions were then tasked with recovering 5% of these outliers to their "original" values. We evaluated the methods using several machine learning models for classification, focusing on specific scenarios due to resource constraints. For the Forest Fire dataset, we selected the Relative Humidity (RH) feature as the outlier and used K-Nearest Neighbors (KNC) for classification. For the PEMS dataset, we targeted the "400017" feature with an Autoencoder for classification. For the UNSW dataset, we used the "dur" feature with Random Forest classification. We compared the four solutions using key metrics, including accuracy, F1 score, precision, and recall, and detailed other classification configurations in the setup Section V-A. The code used for these experiments is publicly available (see footnote 2), and the results presented are the averages of three executions.

The results in Table. VI show that our proposed solution achieved the highest performance across the Forest Fire, UNSW, and PEMS datasets, proving its strength and efficiency and making it the most reliable method among those tested. The FEM algorithm [48] does not achieve the same level of accuracy in predicting missing data because it assumes a specific distribution that might not align well with all datasets, leading to less precise imputations. The NSS method [49] sometimes gives biased results, especially when the data is highly skewed. Similarly, the AMSA-VAE method [50] is great at recovering missing data but does not tackle data imbalance directly, limiting its overall impact in such cases.

Our FID approach takes a different angle, treating missing data and data imbalance as interconnected issues, offering a more unified and consistent way to repair data. It uses fuzzy-based information decomposition to estimate missing values while keeping the original data distribution intact, avoiding unrealistic results. Moreover, the FID method outperformed other solutions by combining data recovery and balancing into one seamless process, enhancing classification performance even when dealing with missing and imbalanced data at the same time. Overall, our proposed method has shown superior results compared to the existing approaches.

VII. FURTHER DISCUSSION

We note that further evaluation of this work is an interesting research direction. *Firstly*, we notice that our method does not predict and recover the exact original value of compromised data, an example is shown in Table VII. Instead, the recovery data values are estimated by the Fuzzy-Based Information Decomposition method which computes the weighting contribution based on all observed data. So, with the method,

TABLE VII
SAMPLE OF RECOVERY VALUES ON FOREST FIRE DATASET.

The Index position	*23*	*27*	*29*	*42*	*93*	*104*
The original data values	47.47	17.95	9.34	-1.57	33.39	16.42
The recovery data values	8.78	16.82	22.51	29.39	37.02	55.02

although the recovery data values are not the same as original data values, it still plays an important role to ensure that compromised data does not affect the AI decision making operations. Therefore, the proposed framework which aims to detect faulty data and mitigate its impact to AI decision making is effective. This key limitation to predict exact original data value/s will be investigated in future.

Secondly, we aim to conduct further experiments to jointly evaluate how the proposed method can classify the imbalanced data samples with missing values, both at the same time. Because it is possible that the data sets used to train AI models are not complete although the models results high accuracy but do not preform well on minority class. *Following this*, a detailed comparison of this proposal with [41] is sought to be conducted. Currently, with limited resources and time we were not able to investigated these aspects. *Also,* one can argue that there is no need of recovering the compromised or missed data (not the data imbalanced issue) because we have solutions available handling a small dataset problem in prediction model by employ artificial data generation approaches [17], however how legitimate the small data is a questionable issue. So further comparison is also needed with recent approaches in line to this argument.

Moreover, the proposed methodology needs further improvements. For example, in our work the approach first identifies data sparsity issues (missing or manipulated sensor readings) at run-time and then fixes those values. Yet, in our further evaluation, we input the data sparsity elements directly to the FID module to estimate them, instead of first automatically identifying them at run-time. Additionally, we injected Gaussian noise to mimic an attacker, however this is a limited scenario of evaluation. The work should be extended to identify the attackers complicated attack strategies to manipulate actuator or sensor values to cause serious consequences in an IoT environment. Therefore, further evaluation is needed.

Further, not in our work but many of the existing works we note that the 'faulty data' is equated to 'outliers'. Although this is acceptable, however it means that the system ignores the case where an adversary manipulates data by replacing valid data with other valid data (non-outliers). The assumption appears to be that attackers will always affect the system in a way that can be detected by the 'compromised data detection' algorithm. Hence, if this were possible and reliable, then the attack does not actually succeed. We emphasize that this is a very interesting issue to explore to further enhance the robustness of the proposed approach or the existing approaches.

Finally, the approach assumes all the attributes in the sample are independent of each other, which may not be true in real-world cases. In counter argument we justify this assumption with a brief a discussion. Note that sensor values in dynamic IoT sensor networks differ significantly from real-world data like biomedical data, DNA sequences, and other time-series datasets due to the inherent independence and variability of the attributes. In IoT environments, attributes such as packet sizes, packet inter-arrival times, and byte frequencies are usually independent of each other, meaning one attribute does not necessarily affect another. For example, slower internet speeds may take longer to transfer the same packet size than faster internet speeds; in this case, there is no strong connection between these two attributes. Therefore, if a method assumes these attributes are dependent and builds a connection, this type of missing data recovery technique may introduce noise. However, since the FID method assumes independence between attributes, it can effectively recover missing values, leading to better performance compared to methods that do not account for this independence. Additionally, sensor spacing is determined not by fixed endpoints but by dynamic model parameters that represent the physical conditions of the environment, such as sensor placement or road characteristics. All of these independence and reliance on dynamic conditions make IoT sensor data unique, requiring specialized methods like FID for accurate analysis. However, we still plan to further improve the FID algorithm by considering the relationships among the attributes. Additionally, conducting a deep study of the relationship between the degree of imbalance and the percentage of missing fault values is an interesting endeavour which we plan to evaluate this in the future.

Recently, the authors in [51] have proposed an explainable trust scoring model that maps the IoT device level evidence into trust scores in a way that produces lower trust scores when devices are under attack. The evaluation of their approach (using two real data sets) shows solid results which becomes the base of investigating unified solutions which are not attack or device-specific, but instead can be generalized across most attack types, devices, and services in smart home IoT. At this stage, in our opinion, we believe that using AI methods the work can be further enhanced and applied in the autonomous trust calculation/enhancement in zero trust IoT architecture domain in which high level of data sparsity is common and their FID approach will complement the solution in [51].

VIII. Summary and Future Work

The paper proposed a method to inject artificial data samples into a data stream that is processed by a machine learning algorithm. If an attacker were to block parts of the data stream, the algorithm might produce incorrect results. The proposed method generates data to fill the missing pieces so that the algorithm can continue to function even in the presence of this adversary. We have proposed a fuzzy logic-based scheme to recover the missed data. We have used three data sets from three different domains to validate the effectiveness of the scheme. Our study showed that as cyber security risks are growing; it is highly expected to incorporate the data sparsity to effectively detect as well as recovers the missed or compromised data values (not only to detect the compromised data values). We have also discussed the genuine limitations of the work in the further discussion section on which we aim to work in future.

References

[1] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of iot: Applications, challenges, and opportunities with china perspective," *IEEE Internet of Things journal*, vol. 1, no. 4, pp. 349–359, 2014.

[2] D.-W. Huang, F. Luo, J. Bi, and M. Sun, "An efficient hybrid ids deployment architecture for multi-hop clustered wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2022.

[3] W. Xu, T. Xiao, J. Zhang, W. Liang, Z. Xu, X. Liu, X. Jia, and S. K. Das, "Minimizing the deployment cost of uavs for delay-sensitive data collection in iot networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 812–825, 2021.

[4] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE communications surveys & tutorials*, vol. 21, no. 2, pp. 1744–1772, 2018.

[5] D. Fooladivanda, Q. Hu, Y. H. Chang, and P. Sauer, "Secure state estimation and control for cyber security of ac microgrids," *arXiv preprint arXiv:1908.05843*, 2019.

[6] K. Sood, S. Yu, D. D. N. Nguyen, Y. Xiang, B. Feng, and X. Zhang, "A tutorial on next generation heterogeneous iot networks and node authentication," *IEEE Internet of Things Magazine*, vol. 4, no. 4, pp. 120–126, 2021.

[7] B. Feng, A. Tian, S. Yu, J. Li, H. Zhou, and H. Zhang, "Efficient cache consistency management for transient iot data in content-centric networking," *IEEE Internet of Things Journal*, 2022.

[8] N. Kaloudi and J. Li, "The ai-based cyber threat landscape: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–34, 2020.

[9] M. Paknezhad, C. P. Ngo, A. A. Winarto, A. Cheong, C. Y. Beh, J. Wu, and H. K. Lee, "Explaining adversarial vulnerability with a data sparsity hypothesis," *Neurocomputing*, 2022.

[10] C. Zhang, X. Costa-Pérez, and P. Patras, "Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms," *IEEE/ACM Transactions on Networking*, 2022.

[11] K. Sood, M. R. Nosouhi, N. Kumar, A. Gaddam, B. Feng, and S. Yu, "Accurate detection of iot sensor behaviors in legitimate, faulty and compromised scenarios," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[12] G. R. Mode, P. Calyam, and K. A. Hoque, "Impact of false data injection attacks on deep learning enabled predictive analytics," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–7.

[13] A. Alsaedi, Z. Tari, R. Mahmud, N. Moustafa, A. N. Mahmood, and A. Anwar, "Usmd: Unsupervised misbehaviour detection for multi-sensor data," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[14] S. Krithivasan, S. Sen, and A. Raghunathan, "Sparsity turns adversarial: Energy and latency attacks on deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 4129–4141, 2020.

[15] B. Feng, H. Zhou, G. Li, Y. Zhang, K. Sood, and S. Yu, "Enabling machine learning with service function chaining for security enhancement at 5g edges," *IEEE Network*, vol. 35, no. 5, pp. 196–201, 2021.

[16] C. Chen, X. Zhao, and M. C. Stamm, "Generative adversarial attacks against deep-learning-based camera model identification," *IEEE Transactions on Information Forensics and Security*, 2019.

[17] M. A. Lateh, A. K. Muda, Z. I. M. Yusof, N. A. Muda, and M. S. Azmi, "Handling a small dataset problem in prediction model by employ artificial data generation approach: A review," in *Journal of Physics: Conference Series*, vol. 892, no. 1. IOP Publishing, 2017, p. 012016.

[18] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, 2019.

[19] A. J. Smola, S. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *International Workshop on Artificial Intelligence and Statistics*. PMLR, 2005, pp. 325–332.

[20] K. Chen, G. Chu, X. Yang, Y. Shi, K. Lei, and M. Deng, "Hseta: A heterogeneous and sparse data learning hybrid framework for estimating time of arrival," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[21] S. Liu, J. Zhang, Y. Xiang, and W. Zhou, "Fuzzy-based information decomposition for incomplete and imbalanced data learning," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1476–1490, 2017.

[22] S. Birnbach and S. Eberz, "Peeves: Physical event verification in smart homes," *Journal of Smart Home Security*, 2019.

[23] C. Fu, Q. Zeng, and X. Du, "{HAWatcher}:{Semantics-Aware} anomaly detection for appified smart homes," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 4223–4240.

[24] S. Birnbach, S. Eberz, and I. Martinovic, "Haunted house: physical smart home event verification in the presence of compromised sensors," *ACM Transactions on Internet of Things*, vol. 3, no. 3, pp. 1–28, 2022.

[25] M. O. Ozmen, R. Song, H. Farrukh, and Z. B. Celik, "Evasion attacks and defenses on smart home physical event verification," in *Proceedings of the NDSS Symposium 2023*. NDSS, 2023.

[26] G. Singh, K. Sood, P. Rajalakshmi, D. D. N. Nguyen, and Y. Xiang, "Evaluating federated learning based intrusion detection scheme for next generation networks," *IEEE Transactions on Network and Service Management*, 2024.

[27] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1119–1130, 2012.

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

[29] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed-attribute data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 1, pp. 110–121, 2010.

[30] X. Ye, I. Esnaola, S. M. Perlaza, and R. F. Harrison, "Stealth data injection attacks with sparsity constraints," *arXiv preprint arXiv:2201.00065*, 2021.

[31] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[32] X. Ye, I. Esnaola, S. M. Perlaza, and R. F. Harrison, "Information theoretic data injection attacks with sparsity constraints," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.

[33] H. T. Reda, A. Anwar, A. Mahmood, and N. Chilamkurti, "Data-driven approach for state prediction and detection of false data injection attacks in smart grid," *Journal of Modern Power Systems and Clean Energy*, 2022.

[34] K.-D. Lu and Z.-G. Wu, "Multi-objective false data injection attacks of cyber-physical power systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2022.

[35] M. Kordos, M. Blachnik, and R. Scherer, "Fuzzy clustering decomposition of genetic algorithm-based instance selection for regression problems," *Information Sciences*, vol. 587, pp. 23–40, 2022.

[36] K. C. Sou, H. Sandberg, and K. H. Johansson, "On the exact solution to a smart grid cyber-security analysis problem," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 856–865, 2013.

[37] M. Pawlicki, M. Choraś, R. Kozik, and W. Hołubowicz, "Missing and incomplete data handling in cybersecurity applications," in *Intelligent Information and Database Systems: 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, April 7–10, 2021, Proceedings 13*. Springer, 2021, pp. 413–426.

[38] L. Yang, M. E. Rajab, A. Shami, and S. Muhaidat, "Enabling automl for zero-touch network security: Use-case driven analysis," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2024.

[39] A. Tharwat and W. Schenck, "Active learning for handling missing data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2024.

[40] A. D. Woods, D. Gerasimova, B. Van Dusen, J. Nissen, S. Bainter, A. Uzdavines, P. E. Davis-Kean, M. Halvorson, K. M. King, J. A. Logan *et al.*, "Best practices for addressing missing data through multiple imputation," *Infant and Child Development*, vol. 33, no. 1, p. e2407, 2024.

[41] Z. Fang, M. Xu, S. Xu, and T. Hu, "A framework for predicting data breach risk: Leveraging dependence to cope with sparsity," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2186–2201, 2021.

[42] K. Sood, M. R. Nosouhi, N. Kumar, A. Gaddam, B. Feng, and S. Yu, "Accurate detection of iot sensor behaviors in legitimate, faulty and compromised scenarios," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 288–300, 2023.

[43] Y. Miao, C. Chen, L. Pan, Q.-L. Han, J. Zhang, and Y. Xiang, "Machine learning–based cyber attacks targeting on controlled information: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–36, 2021.

[44] L. Ren, T. Wang, A. S. Seklouli, H. Zhang, and A. Bouras, "A review on missing values for main challenges and methods," *Information Systems*, p. 102268, 2023.

[45] H. Xu, Z. Sun, Y. Cao, and H. Bilal, "A data-driven approach for intrusion and anomaly detection using automated machine learning for the internet of things," *Soft Computing*, vol. 27, no. 19, pp. 14 469–14 481, 2023.

[46] G. Bovenzi, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, “Network anomaly detection methods in iot environments via deep learning: A fair comparison of performance and robustness,” *Computers & Security*, vol. 128, p. 103167, 2023.

[47] B. M. S. Hasan and A. M. Abdulazeez, “A review of principal component analysis algorithm for dimensionality reduction,” *Journal of Soft Computing and Data Mining*, vol. 2, no. 1, pp. 20–30, 2021.

[48] F. Mouret, A. Hippert-Ferrer, F. Pascal, and J.-Y. Tourneret, “A robust and flexible em algorithm for mixtures of elliptical distributions with missing data,” *IEEE Transactions on Signal Processing*, vol. 71, pp. 1669–1682, 2023.

[49] J. Dou, G. Wei, Y. Song, D. Zhou, and M. Li, “Switching triple-weight-smote in empirical feature space for imbalanced and incomplete data,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 2, pp. 1850–1866, 2024.

[50] L. Chen, Y. Xu, Q.-X. Zhu, and Y.-L. He, “Adaptive multi-head self-attention based supervised vae for industrial soft sensing with missing data,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 3, pp. 3564–3575, 2024.

[51] H. Alsheakh and S. Bhattacharjee, “Towards a unified trust framework for detecting iot device attacks in smart homes,” in *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2020, pp. 613–621.