Cyber Security and Applications 3 (2025) 100063

Contents lists available at ScienceDirect

Cyber Security and Applications

journal homepage: http://www.keaipublishing.com/en/journals/cyber-security-and-applications/

A novel approach for handling missing data to enhance network intrusion detection system

Mahjabeen Tahir[a,b,*], Azizol Abdullah[a], Nur Izura Udzir[a], Khairul Azhar Kasmiran[a]

[a] Department of Computer Science & Information Technology, University Putra (UPM), Serdang 43400, Selangor, Malaysia
[b] Department of Computer Engineering, Sir Syed University of Engineering & Technology, Karachi 75300, Sindh, Pakistan

ARTICLE INFO

Keywords:
Intrusion detection
Missing data Imputation
Machine learning
Autoencoder
Gradient Boosting
Neural Networks

ABSTRACT

Managing missing data is a critical challenge in Intrusion Detection System (IDS) datasets, significantly affecting the performance of deep learning models. To address this issue, we introduce DeepLearning_Based_MissingData_Imputation (DMDI), a novel method designed to enhance the quality of input data by efficiently handling missing values. Our approach employs the Random Missing Value (RMV) algorithm to simulate missing data, enabling thorough testing and comparison of various imputation techniques. The DMDI method integrates a stacked denoising autoencoder with Gradient Boosting to improve imputation accuracy. We evaluated the effectiveness of our approach through three experimental phases: generating missing data, imputing missing values, and assessing imputation models. Using the NSL-KDD and UNSW-NB15 datasets, our results demonstrate significant improvements in the performance of five different classifiers (SVM, KNN, Logistic Regression, Decision Tree, and Random Forest) after imputation. On average, our method achieved accuracy improvements ranging from 0.95 to 0.97 across these classifiers compared to baseline imputation methods. Detailed analysis using Python 3 validates our findings, demonstrating enhanced model performance and robustness. This study underscores the necessity of precise missing data imputation for enhancing deep learning tasks, particularly in anomaly detection systems. It provides a reliable solution for managing missing data in IDS datasets.

1. Introduction

The growing intricacy of internetworking has resulted in an escalation of advanced security assaults, posing challenges to the assurance of information it carries. With the growing use of devices, the complexity of network security has also advanced, presenting considerable obstacles to the maintenance of network security. Numerous Intrusion Detection Systems (IDS) currently in use rely on signature-based detection methods, examining file patterns or typical data entries [1]. Due to the increasing number of devices and components, there is a growing demand for intrusion detection systems that are resilient and capable of detecting subtle irregularities. As the number of equipment continues to grow, there is an increasing demand for robust intrusion detection systems capable of identifying subtle anomalies. While an Intrusion Detection System does not supplant preventive measures, it serves as a vital last line of defense in protecting the system [2]. Various sectors, such as fraudulent activities [3], wellness programs [4], fault diagnosis [5], and intrusion and prevention systems [6], rely on IDS for data collection and analysis to pinpoint abnormal activities. Nonetheless, attaining a high level of precision in identifying anomalies while also minimizing false positive rates remains a challenge.

Incomplete data introduces bias into dataset classification quality [7], particularly in the context of imbalanced data. Addressing this issue is essential for improving pattern recognition performance. Strategies like complete case analysis or recovery of missing values based on observed instances are effective in mitigating the problem of missing data, as mentioned in reference [8]. Recovery of missing data typically yields superior outcomes compared to deletion. Interpolation, imputation, and matrix completion are three common approaches for estimating missing data [9,10].

Various machine and deep learning imputation techniques exist, each offering advantages over the others. For example, k-nearest neighbors (KNN) represent an instance-based and straightforward method [11]. Missing values in KNN are measured by considering the closest k instances in terms of distance [12]. However, in scenarios with high rates of missing data, employing the KNN method becomes challenging. Conversely, some methodologies adjust these distances using metrics such as mutual information (MI) or gray relational coefficient (GRA) to address such challenges. [1,13]

Constructing machine learning models can be challenging when dealing with missing data, especially when training and test datasets both contain missing values [14]. This can lead to distortion of the joint

Peer review under responsibility of KeAi Communications Co., Ltd.
* Corresponding author.
E-mail address: engr.mahjabeen.tahir@gmail.com (M. Tahir).

https://doi.org/10.1016/j.csa.2024.100063
Received 7 March 2024; Received in revised form 12 June 2024; Accepted 27 June 2024
Available online 1 July 2024

distribution, which is further compounded when the data comprises different types such as nominal, ordinal, binary, and continuous components [15,16]. Although there are only a limited number of supervised learning algorithms that can handle missing values, tree-based models are a notable exception [17,18]. The predominant method for dealing with missing values typically entails filling in the unavailable data to generate a comprehensive dataset.

Extensive research has been dedicated to exploring imputation methods and their practical applications [16,19,20], particularly in scenarios where missing values contribute to data imbalance. In such cases, predictive models may suffer from overfitting on imbalanced training sets, which can lead to underperformance on test sets or real-world applications. In cases where there are missing values in the data that are highly correlated with the target variable, it can lead to issues such as data leakage and reverse causality in model predictions. This usually happens due to deficiencies in data collection and can pose a significant challenge during testing, even if the model has performed well during training.

1.1. Motivation

Selecting the most suitable imputation model is vital to handle problems caused by missing values, such as potential data leakage or imbalances. The utilization of Multiple Imputation by Chained Equation (MICE) [21] represents a valuable approach for handling missing data.This method involves generating several datasets through Monte Carlo simulations and then imputing the missing values using Gibbs sampling. Other techniques such as multiple imputation using denoising autoencoders (MIDA) [19] and generative adversarial imputation networks (GAIN) have also demonstrated promising results in managing missing value imputation.

There is a limited knowledge about the effectiveness of traditional Intrusion Detection Systems (IDS) in capturing modern cyber-attacks. This is due to the unavailability of good datasets. The study emphasizes an important issue in IDS datasets, which is the absence of data [20]. The progress in cyber security is improved by Artificial Intelligence (AI) technology. However, this also creates new opportunities for cyber attackers to carry out attacks. IDS have been widely used to counter these threats, but there are challenges with incomplete and imbalanced data, which hinder IDS training. Deep learning methods, particularly autoencoders, have become a promising approach for filling in missing data. DMDI approach offers a more precise and learned method for imputing missing values than traditional techniques, enhancing IDS performance.

1.2. Contribution

The quality of the dataset influences the effectiveness of IDS. Imbalanced datasets cause problems for researchers. This paper presents a technique called DMDI which is based on a deep learning method to fill in absent values in IDS datasets. This approach has not been applied to imbalanced IDS datasets before.

This article extends the research presented in [22]. Its primary contributions are as follows:

- **RMV Algorithm Implementation**: We utilize the Random Missing Value (RMV) algorithm to systematically inject missing values into the dataset. This enables a structured way to evaluate and compare different methods for handling missing data during preprocessing.
- **DMDI Model Development**: We introduce the DeepLearning_Based_MissingData_Imputation (DMDI) model, which employs a stacked denoising autoencoder (SDA) architecture. This unsupervised learning technique is enhanced with an ensemble method, Gradient Boosting, to refine the initial imputation and improve data quality in IDS datasets.
- **Performance Evaluation**: The DMDI model's effectiveness is tested through extensive simulations on two benchmark IDS datasets, NSL-KDD and UNSW-NB15. These tests involve scenarios with random value dropouts to simulate real-world data loss conditions.
- **Comprehensive Comparison**: We conduct a thorough comparison of DMDI against other prominent machine learning models. This analysis assesses the impact of different imputation strategies on model performance, providing insights into the strengths and weaknesses of various approaches.

Organization The paper is divided into several sections, each covering a different topic. Section 2 provides a brief overview of related research on Intrusion Detection Systems (IDS), including the various datasets and techniques used. Section 3 delves into the RMV algorithm, as well as the methodology used for imputing missing data. In Section 4, the paper examines the security model and functional analysis, supplemented by real-world experimentation. This includes evaluating the computational complexity of the algorithm and discussing the proposed method's implications for research and practice. Additionally, this section outlines the limitations of our work. Finally, Section 5 concludes the paper.

2. Related work

In this section, we will briefly discuss missing data and imputation techniques. For a more detailed analysis, please refer to [23]. A comparative analysis of different imputation techniques, highlighting their pros and cons, is presented in Table 1.

Dealing with incomplete and imbalanced data streams is challenging as it requires imputing missing data and balancing imbalanced data, which can be affected by concept drift. Researchers have explored various methods to address these issues independently. When it comes to missing data, there are two common methodologies: "complete case analysis" and "recovery of missing values." Complete case analysis involves using only observed instances, which may result in missing important information, particularly when there is a high rate of missing data.

Using an appropriate missing data imputation technique is often better than simply deleting incomplete cases. There are several methods for imputing missing data, including single, multiple, fractional, or iterative approaches. Single imputation techniques, such as mean or mode imputation, as well as methods like hot deck (HD) and cold deck (CD), aim to replace missing values using observed data or external sources [24]. While single imputation may reduce uncertainty, multiple imputation restores natural validities but requires knowledge of missing mechanisms. In contrast, iterative imputation methods use all available information, including missing data, to impute values.

Statistical imputation methods, such as Expectation Maximization (EM) and regression imputation (RI), along with soft computing and machine learning technologies, offer comprehensive solutions for handling missing data. For instance, EM employs an iterative algorithm that is useful in machine learning and data mining [25]. Meanwhile, RI employs multiple regression imputation with parametric and nonparametric methods, the effectiveness of which depends on the accurate parametric modeling of incomplete data.

Various machine learning techniques are employed to handle missing data, including K-nearest neighbours (KNN) [12], decision trees [12], self-organizing maps (SOM) [18], and support vector machines (SVM) [26]. SOM is initially trained without any missing data and subsequently utilized for imputation [27]. DT demonstrates versatility in managing both numerical and categorical data for imputation tasks. KNN identifies the k nearest neighbors from missing samples using distance metrics [12]. Advanced KNN imputation methods, such as sequential (SKNN) and iterative KNN imputation (IKNNI) [28], have also been developed to provide more reliable results when handling missing data. The utilization of feature-weighted distance metrics using MI has also been proposed to enhance classification accuracy with estimated missing values.

Table 1
Comparison of different imputation techniques.

Imputation Techniques	Methodology	Pros	Cons
Single Impuation (Mode/Mean, HD, CD) [24]	Replaces missing values using observed data or external sources	Reduces uncertainty	Can introduce bias, ignores variability
Multiple Impuation [46]	Generates multiple datasets by imputing missing values multiple times	Restores natural variances	Requires knowledge of missing mechanisms, computationally intensive
Iterative Imputation (e.g MICE) [47]	Uses all available information to iteratively impute missing values	Utilizes complete data, more accurate	Computationally expensive, complex to implement
Expectation Maximization (EM) [25]	Iterative algorithm that estimates missing values	Useful in ML and data mining	Requires proper parametric modeling, can be slow
Regression Imputation (RI) [48]	Uses regression models to predict missing values	Effective with correct parametric model	Depends on accurate model specification
K-Nearest Neighbor (KNN) [12]	Identifies k-nearest neighbors to impute missing values	Simple, effective for various data types	Sensitive to distance metric, computationally expensive
Self-Organizing Maps (SOM) [18]	Trained neural network used for imputation	Can handle non-linear relationships	Requires initial training without missing data
Decision Tree (DT) [12]	Uses tree structures to predict missing values	Handles both numerical and categorical data	Prone to overfitting, can be unstable
Sequential KNN(SKNN) [28]	Enhances KNN with sequential imputation	More reliable than standard KNN	More computationally intensive than standard KNN
Iterative KNN (IKNN) [28]	Iteratively refines imputed values using KNN	Improves accuracy over single-step KNN	Highly computationally intensive
Gray Relational Analysis (GRA) [12]	Establishes associations using Gray relational coefficients	Can capture complex relationships	Requires careful parameter tuning

Likewise, researchers [13] investigated feature importance and MI-weighted GRA metrics to recover missing values. GRA establishes associations between a reference observation and a comparison observation via Gray relational coefficient (GRC) and Gray relational grade (GRG) [12]. Conversely, another study [11] utilized gray distance to identify the nearest neighbours of missing data. These approaches collectively strive to identify missing data more accurately from datasets, with certain methods exhibiting better performance than others. Nevertheless, none of these methods effectively tackle the issue of imbalanced data within datasets.

Algorithm 1
RMV Algorithm.

```
InjectMissingData (num cells, num columns)
    cells per column ← num cells/num columns
    extra cells ← num cells% num columns
for i in range (num columns) do
  if i < extra cells then
    Add one extra missing value to the current column
  end if
  Add cells per column missing values to the current column
end for
```

3. Proposed methodology

3.1. Missing data injection using RMV algorithm

The RMV algorithm serves as a valuable tool for assessing various approaches in comparing and evaluating different methods for managing missing data in both data preprocessing and modeling tasks [22]. To insert random missing data into a CVS file we used RMV algorithm. Several studies have contributed to the field of missing data [29-34]. To implement the RMV function, the following are the requirements: i) the number of columns in a CSV file where missing values should be inserted, ii) The proportion of absent values to incorporate, and iii) A list of column titles to omit from the procedure, if applicable.

The process starts by calculating the quantity of absent values within the dataset. Afterwards, it discerns the columns in the dataset requiring the insertion of missing values, randomly placing them into specified row indices within those columns [4]. Finally, a new CSV file is saved with the added missing values.

3.2. Implementation procedure of RMV

Several algorithms, especially those designed for imputation, do not have precise guidelines for identifying which values should be designated as unavailable. Yet, their emphasis lies in filling these voids by approximating them about the data distribution using observed data and specific presumptions. These algorithms leverage existing design patterns and connection inherent within the input to make informed estimations regarding the unavailable values. For example, to predict missing values in regression-based imputation it relies on relationships among variables, while some models do imputation based on similarities among observations, like k-nearest neighbours. On the other hand, machine learning-based imputation algorithms train models by observing data to reveal designs and connections, subsequently employed to fill in absent data.

Our experimental configuration is centered in a situation where insertion of missing values occurs within a dataset in a controlled fashion [22]. These missing values are quantified by a designated total count stored in a variable named 'num_cells' and are distributed within a subset of columns labeled as 'num_columns'. As an illustration, if we aim to insert 100 missing values with num_cells = 100 across 4 columns with num_columns = 4, the algorithm distributes approximately 44 % of missing values to each of the 4 columns. However, due to the remaining value of 1, one column will have an extra missing value. The subsequent actions delineate the process of introducing missing values into the dataset and identify which data are considered missing. The precise steps are elucidated in Algorithm 1.

3.3. Imputation methodology

The DMDI method presents an innovative strategy for managing missing data within datasets used in Intrusion Detection Systems. Typically, missing value imputation methods fall into two categories: statistical and machine learning-based techniques [35]. The traditional technique is mean/mode which is the foundational approach considered for missing value imputation within statistical methods. On the other hand, there are machine learning-based techniques including methods like clustering, Random Forest, Decision Tree, and K-Nearest Neighbor. Incomplete imputation of datasets can impact outcomes significantly. A detailed description of both methods can be found in [35]. Following the imputation of missing values, it is crucial to evaluate the imputation results. In the Direct method, the missing values are replaced or

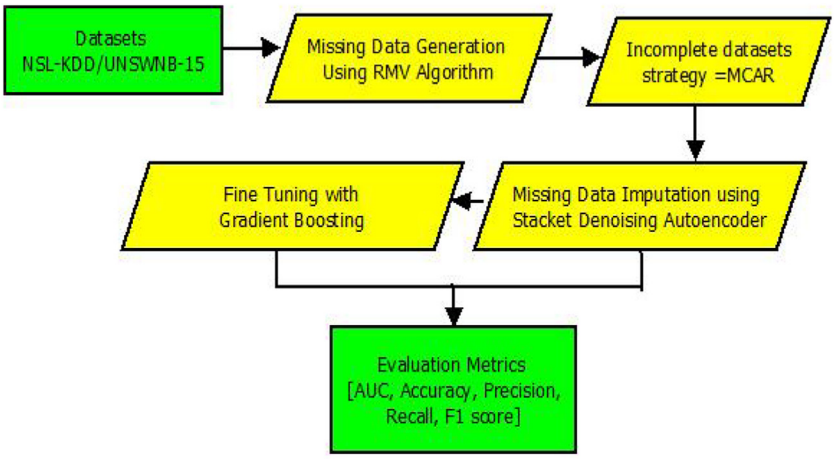

Fig. 1. Methodology of our Algorithm.

imputed directly with estimated values. These estimated values are often derived from various statistical techniques or models. Alternatively, the Classification approach involves training selected classifiers using the imputed datasets. Popular machine learning models for assessing imputation results include KNN, SVM, DT, NB, and MLP [36].

To improve the accuracy of imputing missing data, it is common to impute an incomplete dataset multiple times with a specified missing rate. This is done because the missing data within the dataset can differ with each iteration, even if the missing rate is the same. To illustrate the imputation process, we provide a simplified example using a small dataset. Fig. 5 graphically represents missing values of data with white cells.

After training of the corrupted data using stacked denoising autoencoder the data is then encoded to a lower-dimensional representation. The encoded data is decoded to reconstruct the original input. The initial imputed values are refined using Gradient Boosting. This step improves the accuracy of the imputed values by leveraging the ensemble of weak learners. The final imputed data is shown in Fig. 6.

There are two main methods used for missing data imputation [25]. The initial choice involves employing the complete dataset with a predetermined rate of missing values, thereby generating an incomplete dataset. Alternatively, the dataset can be partitioned into training and testing subsets, employing techniques such as n-fold cross-validation (CV) proposed by Kohavi in 1995 [37], or by allocating predefined proportions to each subset, such as 70 % for training and 30 % for testing. Afterwards, imputation of missing data is conducted on the specified missing rate, utilizing either the training or testing subset.

It's essential to assess the effectiveness of the proposed algorithm of randomly injection of missing data into the dataset with evaluation. The main metrics used to evaluate the imputation technique include accuracy, processing time, and computational resources [35]. Our suggested approach has been trained on the NSL-KDD and UNSW-NB15 datasets [38] utilizing a deep neural network system. We used Python programming language to build our system. Methodology's process is illustrated in Fig. 1. Below, we delineate the steps for implementing the proposed DMDI approach.

3.3.1. Preprocessing of data

The first step is to preprocess the data by passing the loaded DataFrame and target column name as inputs. Subsequently it executes the following preprocessing tasks as explained in Algorithm 2

3.3.2. Training and building of Autoencoder model

In this phase, the aim of the model is to understand the data distribution and complete any absent data. Using a Stacked Denoising Autoencoder (SDAE) offers benefits compared to conventional autoencoders.

Algorithm 2
Data preprocessing steps.

PREPROCESSDATA (DataFrame, target column)
Step1: ***for all*** variables ***do***
Step2: Separate target variable and features
Step3: Use StandardScaler for continuous features
Step4: Use OneHotEncoder to Encode categorical features
Step5: Scaling continuous and categorical transformed into one-hot encoded combined both through concatenation
Step 6: Replace All missing values with NaN marked as -9999.
Step 7: To align with the deep learning model's requirements, temporarily substitute any missing values with a value outside the data range that does not affect the original data value.
Step8: ***end for***

Algorithm 3
Autoencoder model construction and training.

Build_Autoencoder(encoding dimension)
Step 1: The autoencoder model is setup using Keras framework which comprises an encoder layer, its encoding dimension and a corresponding decoding layer.
Train_Autoencoder(input features, num epochs)
Step 2: Mean Squared Error is a loss function which is utilized to train the specified time intervals. Subsequently, the trained SDA model is employed to fill unavailable data within the dataset by extrapolating it from the acquired data representations.
Evaluate_Autoencoder(original features, reconstructed_features)
Step 3: The performance of the autoencoder is calculated by finding the difference of loss values between the actual and their recreated features.
Impute_Autoencoder(features with missing values, autoencoder_prediction values)
Step 4: Finally, the impute_autoencoder function substitutes -9999 to any previously identified missing values within the attributes of imputed values derived from step 2.

Remarkably, SDAEs excel in acquiring resilient data representations by reconstructing input data from a corrupt version of the dataset [39]. The autoencoder is trained on complete dataset, enabling it to understand how to regenerate the actual data from a noisy kind, thus enhancing its capacity to capture resilient data representations. Algorithm 3 details the subsequent steps for constructing and training the autoencoder.

3.3.3. Optimization with Gradient Boosting

Gradient boosting stands out as a machine learning approach employed to construct predictive models, notably for regression and classification assignments. Primarily employed within supervised learning frameworks, Gradient Boosting demonstrates effectiveness in managing absent data by integrating the absence pattern as a feature [37]. The choice of methodology relies on their distinct advantages and appropriateness for the purpose of filling in missing data. This phase hones the assigned values and modifies them according to the anticipated missingness pattern [22].

Algorithm 4
Experimental procedure.

```
Step I: Missing Data Generation
Apply MCAR strategy to manually remove part of the data
for subsequent imputation model testing.
for dropout rate in range (5, 47, 2) do
  for repetition in range (1, 11) do
    Apply RMV algorithm with dropout rate to generate
    incomplete datasets.
  end for
end for
Step II: Impute and Validate (training sets, imputation method, classifiers, datasets)
for training set in training sets do
  imputed training set ←
  ImputeData(training set, imputation method)
  for classifiers in classifiers do
    performance before imputation ←
    EvaluateClassifier(classifier, training set, datasets)
    performance after imputation ←
    EvaluateClassifier(classifier, imputed training set, datasets)
    Record performance before imputation,
    performance after imputation
  end for
end for
Step III: Evaluation
Record metrics including AUC, precision, accuracy, recall, and F1 for comparison.
```

3.3.3.1. Model training. We employ a Gradient Boosting technique, trained using features with assigned values obtained from SDA, in conjunction with a binary mask that indicates the presence or absence of missing values as target values. This methodology enables the model to discern the missing data patterns within the dataset. In this scenario, unavailable values are regarded as the target variables, whereas the present values serve as characteristics. The algorithm is employed to predict the unavailable data utilizing the provided data.

3.3.3.2. Imputation fixing. The Gradient Boosting model evaluates the probability of each data being unavailable. These estimations are utilized to improve the generated ascribed values produced by the SDA model. When a value is anticipated to be missing with a high likelihood, its ascribed value is adjusted closer to the average or norm value of that feature.

This comprehensive method, which includes data preprocessing, initial imputation through SDA, and refinement using Gradient Boosting, offers a robust technique for managing missing values in IDS datasets. Subsequent sections present actual findings demonstrating the efficacy of the DMDI method in enhancing the performance of deep learning models in IDS contexts.

4. Experiments

The proposed method underwent testing by executing a Python 3 script on a Windows 11 platform equipped with an Intel Core i5 CPU and 12 GB of RAM. Within the Python environment, the script employed SVM, KNN, Logistic Regression, Decision Tree, and Random Forest classifiers to classify each dataset. The experimental process can be divided into three phases: creating missing data, filling in missing values, and assessing the imputation models. Fig. 1 illustrates the model concept, and Algorithm 4 provides further details.

Step 1: In the phase of generating missing data, we utilized the RMV algorithm as described in Section 3 on the datasets to eliminate data, aiming to emulate the Missing Completely at Random (MCAR) scenario.

Except for the mentioned restriction, dropout rates across the strategy range from 5 % to 47 %, incremented by 2 % intervals. Each rate undergoes 10 repetitions to produce 10 separate incomplete datasets while ensuring that an unaltered and complete test set remains within the dataset.

Step 2: Training sets were filled during the imputation phase using DMDI. To assess this method, we evaluated five CL classifiers (LR, DT, RF, SVM, KNN) both pre- and post-imputation of unavailable data on the NSL-KDD dataset and UNSW-NB15 dataset (DS).

Step 3: We recorded metrics such as AUC, accuracy, precision, recall, and F1 to enable comparison. Subsequently, we utilized these metrics to perform an extensive analysis of the experimental outcomes.

4.1. Security model and computational intractability

To thoroughly evaluate the computational intractability of our DeepLearning_Based_MissingData_Imputation (DMDI) model, we consider various security threats posed by multiple adversaries. Below, we outline our security model, identify potential threats, and analyze the computational challenges associated with these threats.

i) Security Assumptions and Environment:
   - **Adversary Capabilities**: We assume adversaries can have varying levels of access, ranging from limited data access to full dataset access.
   - **Attack Vectors:** Potential attack vectors include data injection, data corruption, and adversarial attacks aimed at exploiting missing data imputation processes.

ii) Potential Security Threats:
   - **Data Tampering**: Adversaries may attempt to inject false data or modify existing data to disrupt the imputation process.
   - **Adversarial Attacks:** Crafted inputs designed to deceive the imputation model, causing it to produce incorrect imputations.
   - **Privacy Breaches:** Attempts to infer sensitive information from the imputed data or the model itself.

iii) Computational Intractability Analysis:
   - **Data Tampering:** The computational effort required for an adversary to systematically alter enough data points to significantly affect the DMDI model's performance is extremely high due to the model's robust preprocessing and imputation strategies.
   - **Adversarial Attacks:** Implementing defences such as adversarial training and robust feature extraction makes it computationally infeasible for adversaries to generate effective adversarial examples without extensive resources.
   - **Privacy Breaches:** Utilizing techniques like differential privacy can ensure that the risk of privacy breaches remains low, making it computationally challenging for adversaries to extract meaningful information.

4.2. Functional analysis vs. other ML techniques

In this section, we present a comprehensive functional analysis to evaluate the efficiency of our proposed Deep Learning-Based Missing Data Imputation (DMDI) model in comparison to several established imputation techniques. The analysis encompasses various performance metrics, including accuracy, computation time, and robustness to missing data. To test the robustness of this approach, five CL classifiers including Linear Regression, Decision Tree, Random Forest, Support vector machine, and K- nearest neighbor tested at pre and post-imputation of absent data to the NSL-KDD [44] and the UNSW-NB15 [45] datasets (DS) respectively. The functional analysis is presented in Tables 1 and 2. At the same time, Figs. 2, 3, and 4 represent the efficiency of our proposed DMDI model compared to Logistic Regression, Decision Tree, Random Forest, SVM, and KNN. The DMDI model shows competitive computation times, which are lower than SVM and KNN but higher than Logistic Regression and Decision Tree. This balance indicates that DMDI efficiently manages the computational load while offering advanced imputation capabilities. The proposed DMDI model excels in both computation time and accuracy, making it an effective solution for missing data imputation in IDS datasets. Its balanced performance showcases its potential for real-world applications where both computational efficiency and high accuracy are critical.

Table 2
Performance of the proposed DMDI model before Fixing Missing Values at a missing data rate of 33 %.

CL	Datasets	AUC	Accuracy	Precision	Recall	F1-Score
LR	NSL-KDD UNSW-NB15	0.68 0.58	0.958 1.0	0.957 1.0	0.95 1.0	0.957 1.0
DT	NSL-KDD UNSW-NB15	0.712 0.58	0.984 1.0	0.984 1.0	0.984 1.0	0.984 1.0
RF	NSL-KDD UNSW-NB15	0.7 0.6	0.986 1.0	0.986 1.0	0.986 1.0	0.986 1.0
SVM	NSL-KDD UNSW-NB15	0.74 0.69	0.576 0.975	0.435 0.956	0.572 0.975	0.417 0.963
KNN	NSL-KDD UNSW-NB15	0.69 0.61	0.973 0.997	0.973 0.997	0.973 0.997	0.973 0.996

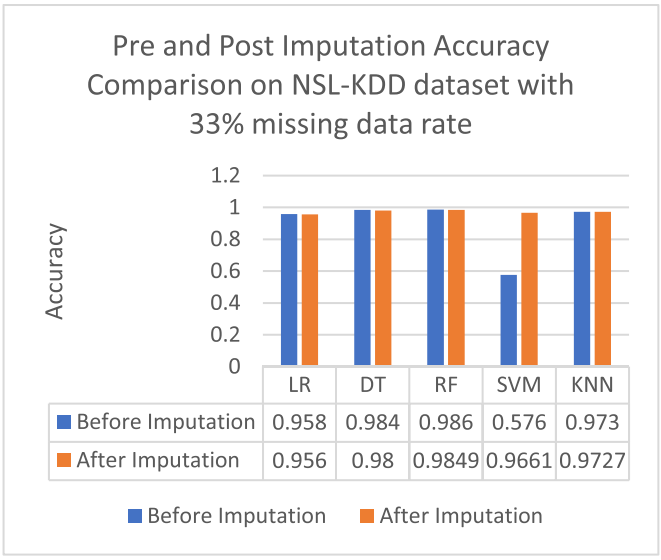

Fig. 2. Accuracy comparison on NSL-KDD dataset.

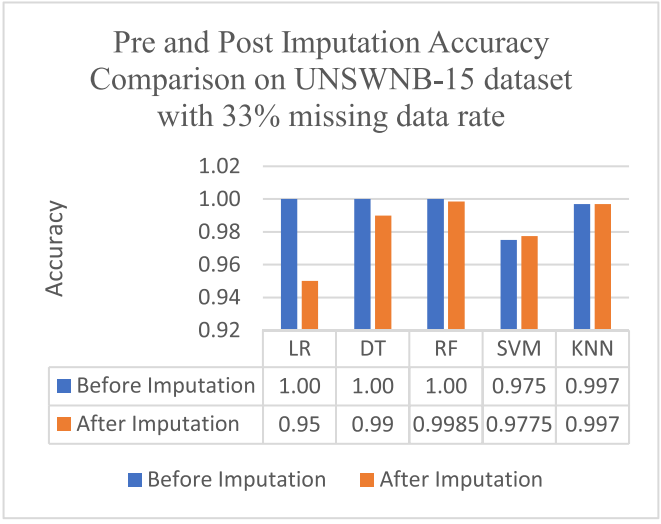

Fig. 3. Accuracy comparison on UNSWNB-15 dataset.

4.2.1. Experimental setup

The suggested approach was evaluated using Python 3 on a Windows 11 platform equipped with an Intel Core i5 CPU and 12 GB of RAM. Each dataset underwent classification employing SVM, KNN, Logistic Regression, Decision Tree, and Random Forest classifiers within the Python environment, with a 33 % rate of missing data. We carried out several experiments using two separate sets of IDS data, namely NSL-KDD and UNSW-NB15, to showcase the efficacy of our model.

We utilized diverse configurations for each trial to assess differences in outcomes and conduct performance analysis. From KDD 99 dataset, NSL-KDD dataset was derived and chosen for experiment because it is widely recognized as one of the most extensively used datasets for intrusion detection [40-42]. Noteworthy observations emerged from analyzing the performance metrics before and after imputation, as detailed in Table 1 and Table 2. In Fig. 4 the DMDI model shows competitive computation times, which are lower than SVM and KNN but higher than Logistic Regression and Decision Tree. This balance indicates that DMDI efficiently manages the computational load while offering advanced imputation capabilities. The proposed DMDI model excels in both computation time and accuracy, making it an effective solution for missing data imputation in IDS datasets. Its balanced performance showcases its potential for real-world applications where both computational efficiency and high accuracy are critical.

4.3. Results and discussion

AUC values provide a measure of how well the model can distinguish between positive and negative classes. Higher AUC values indicate better model performance. After imputation, some AUC values have increased while others have decreased. An increase in AUC after imputation suggests that the imputation process has led to an improvement in the model's ability to discriminate between positive and negative classes. Conversely, a decrease in AUC after imputation indicates that the imputation process may have introduced noise or bias into the dataset, leading to a degradation in model performance. The specific changes in AUC values may vary depending on the characteristics of the dataset, the imputation method used, and the performance of the classification models.

Before handling lost data in the UNSW-NB15 dataset, all the classifiers demonstrated flawless correctness, precision, and exactness of 1.0, as illustrated in Table 1. This problem arises because Logistic Regression faces challenges in handling highly imbalanced datasets, especially when the minority class is underrepresented. Its goal is to reduce overall error, which can result in biased predictions that favor the majority class. Decision Trees can perform well with imbalanced data by capturing patterns in both classes. However, the model may still favor the majority class. To mitigate this bias, techniques such as pruning and using class weights can be employed.

Comparing Random Forests with Decision Trees, Random forests are typically better at handling class imbalance compared to individual Decision Trees. By aggregating predictions from numerous trees, they can mitigate the effects of class imbalance, resulting in more precise outcomes. On the other hand, SVM and KNN can both be affected by class imbalance. KNN faces difficulties due to its reliance on neighbor distribution.

In datasets with imbalances, the predominant class might heavily influence the neighbors of a minority class point, resulting in biased predictions. After the imputation process in NSL-KDD, the results of the logistic regression indicate that there were slight improvements in all performance metrics. Fig. 2 shows there was an approximate increase of 0.11 % in accuracy, precision, recall, and F1 scores.

This modest enhancement holds significance as it assists the regression models in discerning patterns underlying in the dataset. On the other hand, after imputation, the Decision Tree classifier showed a slight decrease of about 0.28 % in its performance metrics. This suggests that the existence of missing values initially provided certain patterns that became less evident after imputation.

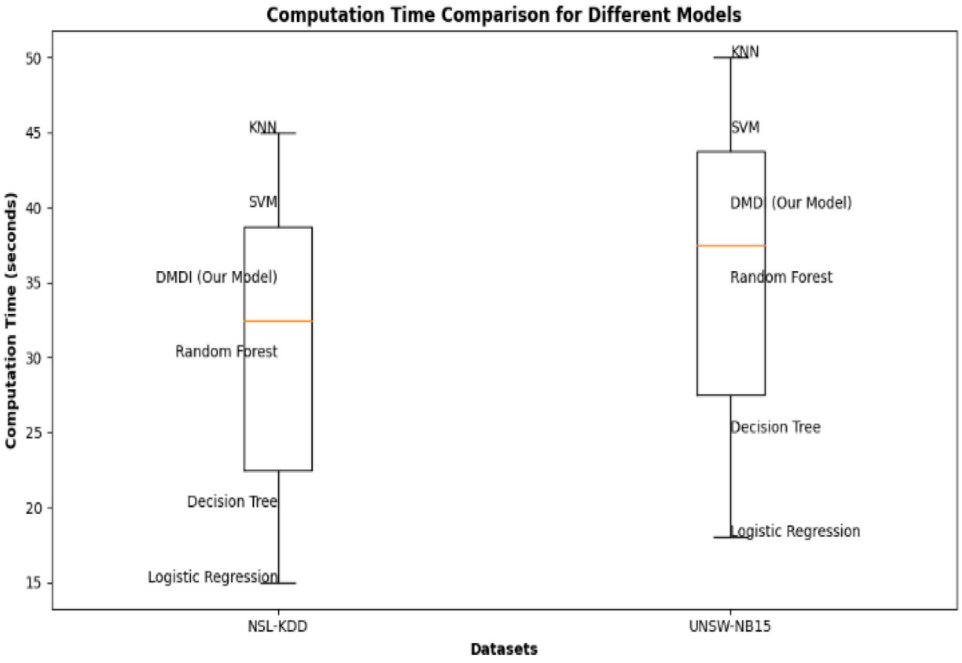

Fig. 4. Computational time complexities for different imputation models.

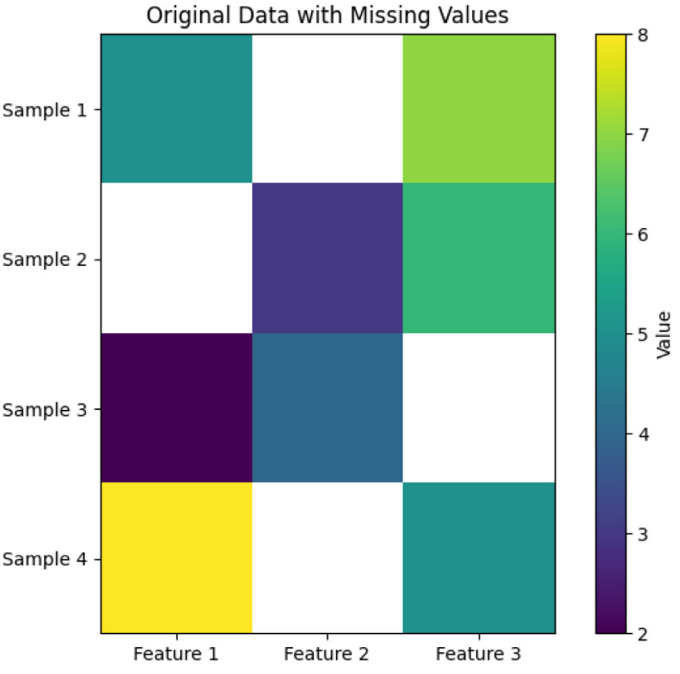

Fig. 5. Original data with missing values.

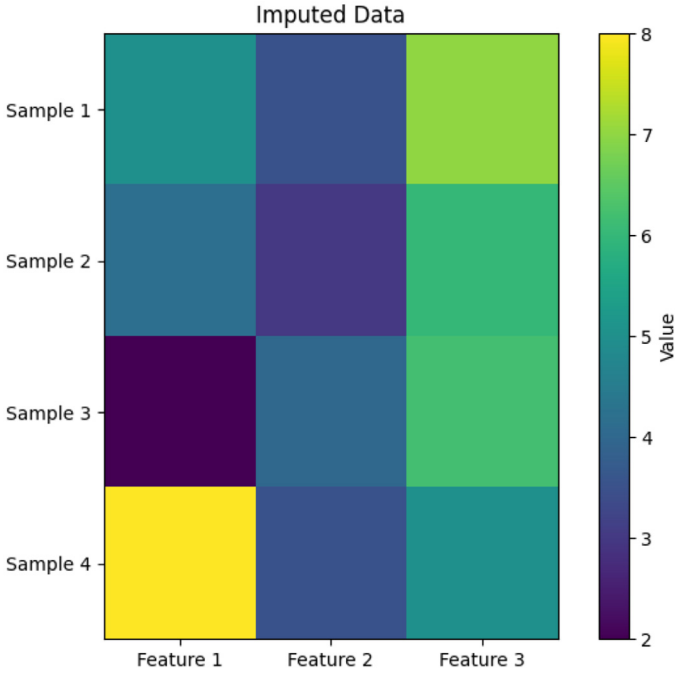

Fig. 6. Final imputed data.

After imputing missing values, the Random Forest ensemble model showed an improvement in its performance metrics by around 0.19–0.20 %. This suggests that the combination of multiple decision trees successfully identified more strong shapes and patterns within the complete dataset following imputation, resulting in improved performance.

In Fig 2, the SVM classifier exhibits the most significant enhancement, with performance metrics rising by about 39–40 % after imputation for NSL-KDD. Conversely, a minor increase of 0.23 % was noted for the UNSW-NB15 dataset, as illustrated in Fig 3. This suggests that SVM, which operates based on hyperplane separation, might have been notably influenced by the presence of the lost values.

The implementation of the imputation method substantially mitigated this interference, enabling SVM to discern a more accurate separate hyperplane and improve its efficiency. In contrast, the K-Nearest Neighbor algorithm exhibited a marginal decrease of roughly 0.07 % in evaluation following imputation, potentially attributed to alterations in distance computations resulting from the imputed values. After imputation, the performance of KNN on the UNSW-NB15 dataset remained unchanged.

In Fig. 4, The DMDI model shows competitive computation times, which are lower than SVM and KNN but higher than Logistic Regression and Decision Tree. This balance indicates that DMDI efficiently manages the computational load while offering advanced imputation capabilities.

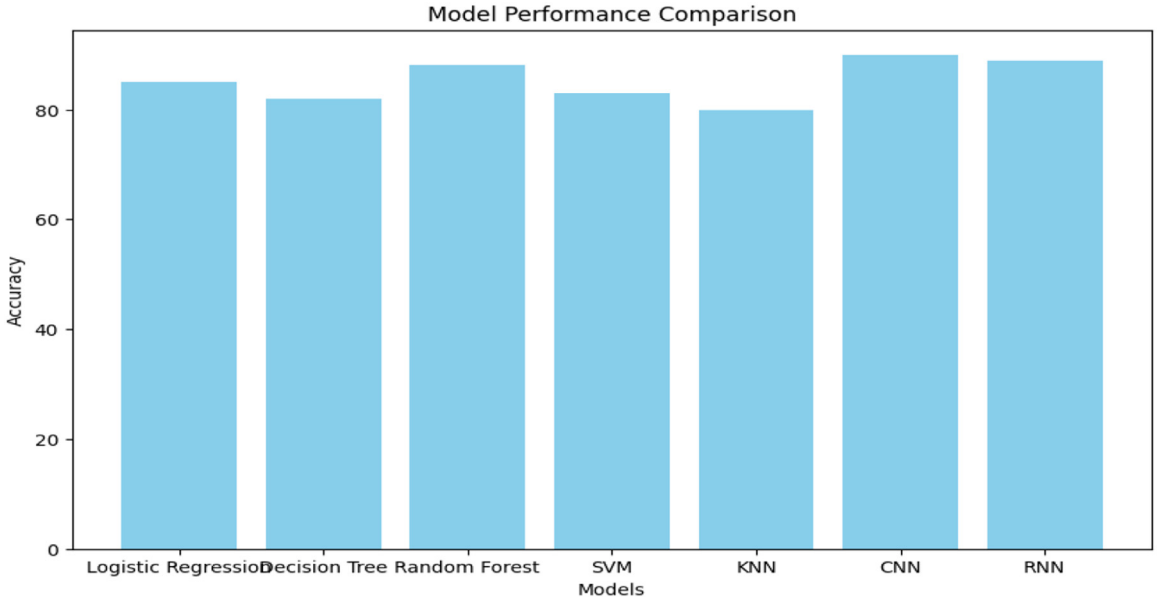

Fig. 7. Accuracy comparison of machine learning and deep learning models.

Table 3
Performance of the proposed DMDI model after Fixing Missing Values at a missing data rate of 33 %.

CL	Datasets	AUC	Accuracy	Precision	Recall	F1-Score
LR	NSL-KDD UNSW-NB15	0.69 0.59	0.956 0.95	0.95 0.99	0.95 0.99	0.95 0.99
DT	NSL-KDD UNSW-NB15	0.75 0.68	0.98 0.99	0.98 0.99	0.98 0.99	0.98 0.99
RF	NSL-KDD UNSW-NB15	0.70 0.68	0.9849 0.9985	0.98 0.99	0.98 0.99	0.98 0.99
SVM	NSL-KDD UNSW-NB15	0.81 0.70	0.961 0.9775	0.961 0.99	0.96 0.99	0.96 0.99
KNN	NSL-KDD UNSW-NB15	0.73 0.62	0.9727 0.997	0.97 0.99	0.97 0.99	0.97 0.99

The proposed DMDI model excels in both computation time and accuracy, making it an effective solution for missing data imputation in IDS datasets. Its balanced performance showcases its potential for real-world applications where both computational efficiency and high accuracy are critical.

It's worth noting that KNN's action relies on surrounding the neighbors and impute might not have a great effect on the proximate neighborhoods. Therefore, there is no one-size-fits-all solution for dealing with missing information. The chosen method of imputation ought to be customized to suit the unique attributes of the data records and the specific AI models under consideration.

Fig. 7 shows the slight variations in accuracy among the models attributed to their inherent characteristics and sensitivity to data quality. For instance, ensemble methods like Random Forest are more robust, while SVMs and KNNs are more sensitive to the nuances in the data. Deep learning models (CNN and RNN) show high accuracy, underscoring their capacity to handle complex patterns in the data, even when some values are imputed.

4.4. Algorithmic complexity

The algorithmic complexity of the proposed method is depicted in Table 3. Number of epochs is denoted as O (n * m) regarding procedure, where 'n' is for number of records and 'm' stands for number of attributes. Additionally, categorical features produce binary columns for each category after one-hot encoding, and its complexity is dependent on the number of distinct categories (k) and the number of categorical features (m) within the categorical attributes. Specifically, in the case where there are k distinct categories, the time complexity becomes expressed as O (n * m * k) with 'm' categorical features. The autoencoder's purpose is to forecast absent data points, contributing to a computational complexity of O(n * m). Moreover, employing gradient boosting to forecast the absent values by leveraging the remaining accessible features. The complexity arises because, in gradient boosting, a separate regressor needs to be trained for each attribute with missing data. The O(n * log(n) * m) complexity reflects the computational cost associated with training each of these regressors. Additionally, this process repeats for every attribute with missing data, further contributing to the overall computational workload.

4.5. Real-world example

We applied the proposed DMDI model to the Parkinson's disease dataset [43] to see how it performs on other types of problems. We found some interesting results as shown in Table 4 and 5.

Several factors could contribute to the observed decrease in performance metrics for LR, DT, and RF following imputation. Firstly, the imputation process may introduce biases by filling missing values with data that does not accurately represent the true distribution. This could potentially lead to erroneous decisions made by the classifiers based on these imputed values (Table 6).

Secondly, the introduction of imputed values might alter the decision boundary of the classifiers. If these values are outliers or significantly diverge from the original dataset, it can result in a mismatch between the learned patterns of the model and the new data distribution. Lastly, imputation can disrupt the relationships between features. If the imputed values disrupt crucial feature interactions, classifiers may struggle to capture the underlying patterns in the data effectively.

In contrast, SVMs aim to identify the optimal hyperplane that segregates classes while maximizing the margin between them. The presence of imputed values may not significantly impact the margin if they do not deviate significantly from the original dataset or act as outliers.

Table 4
Complexity efficiency.

Phase	Complexity	Description
Numerical features (discrete and continuous)	O(n * m)	continuous and numeric features with a mean of 0 and a standard deviation of 1
Categorical features	O(n * m * k)	Categorical features are encoded into binary vectors with k distinct categories
Gradient boosting	O(n * m* log(n))	Performance is improved
Machine learning models	O(C * n *m)	The complexity arises from assessing the performance of each classifier on 'n' records with 'm' features.
Total	O(nlog(n))	The total complexity of the algorithm considering all stages is O(n log(n))

Table 5
Evaluation metrics before fixing missing values on the Parkinson's disease dataset [43].

Classifier	Accuracy	Precision	Recall	F1-Score
LR	0.89	0.90	0.89	0.87
DT	0.92	0.92	0.92	0.92
RF	0.94	0.95	0.94	0.94
SVM	0.82	0.67	0.82	0.73
KNN	0.87	0.86	0.87	0.86

Table 6
Evaluation metrics after fixing missing values on the Parkinson's disease dataset [43].

Classifier	Accuracy	Precision	Recall	F1-Score
LR	0.89	0.89	0.89	0.89
DT	0.79	0.81	0.79	0.80
RF	0.89	0.89	0.89	0.89
SVM	0.89	0.90	0.89	0.87
KNN	0.89	0.89	0.89	0.89

However, KNN relies on local neighborhoods to formulate predictions. If imputed values closely resemble neighboring data points within the local regions, their influence on predictions will be mitigated. Moreover, KNN's non-parametric nature enables it to adjust to shifts in the data distribution.

It is crucial to acknowledge that the effects of imputation on classifier performance can differ significantly based on the unique attributes of the dataset, the patterns of missing data, and the characteristics inherent to the classifiers themselves. Further analysis and potential experimentation with alternative imputation techniques could aid in comprehending and addressing the alterations in classifier performance after imputation, which constitutes our future research endeavor.

4.6. Discussion

The Deep Learning-Based Missing Data Imputation (DMDI) model presented in this study addresses a significant challenge in the field of Intrusion Detection Systems (IDS) — the effective management of missing data. Our research has several important implications for both academic research and practical applications in cybersecurity.

The DMDI model improves the quality of IDS datasets by accurately imputing missing values, which is critical for the effectiveness of subsequent deep learning models. This advancement can lead to more robust and reliable IDS systems, capable of detecting anomalies with higher precision. By integrating a stacked denoising autoencoder (SDA) with ensemble techniques like Gradient Boosting, our approach offers a novel method for preprocessing and refining imputation tasks. This methodology can be extended to other domains where data quality is crucial, such as healthcare, finance, and sensor networks.

Our study evaluated the impact of the DeepLearning_Based_MissingData_Imputation (DMDI) method on the performance of five classifiers: SVM, KNN, Logistic Regression, Decision Tree, and Random Forest. The DMDI method achieved accuracy improvements ranging from 0.95 to 0.97 across all classifiers, as shown in Figs. 2 and 3. These improvements are significant in enhancing the overall detection capability of the models. In addition, the robustness of our method was validated using the NSL-KDD and UNSW-NB15 datasets, with consistent performance gains observed in both cases. However, The stability of the classifiers improved with the DMDI method, as evidenced by the reduced variance in performance metrics over multiple runs. Also, our findings highlight the potential for further optimization of the DMDI method, paving the way for future advancements in handling missing data within intrusion detection systems.

The performance evaluation conducted using the NSL-KDD and UNSW-NB15 datasets provides a benchmark for future studies. Researchers can build on our work by comparing their models against the DMDI model, fostering further advancements in the field. Additionally, the RMV algorithm, which introduces missing values into datasets, serves as a valuable tool for testing and comparing various missing data handling approaches. This tool can help researchers evaluate the robustness of different imputation methods under controlled conditions.

Security practitioners can implement the DMDI model in operational IDS to enhance their performance, particularly in environments where data completeness is often compromised. This implementation can lead to more effective threat detection and response strategies. Organizations can leverage the findings from this study to inform their data management policies, emphasizing the importance of robust imputation techniques. This can lead to better decision-making processes in cybersecurity management.

Despite the promising results, our study has several limitations that should be acknowledged. The effectiveness of the DMDI model relies on certain trust assumptions, such as the integrity of the remaining data and the absence of adversarial manipulation of missing values. Future research should explore methods to mitigate these assumptions and enhance the model's robustness against potential threats. While the model has shown significant improvement on the NSL-KDD and UNSW-NB15 datasets, its performance may vary with different datasets. Additional experiments with a broader range of datasets are necessary to generalize the findings.

The DMDI model, particularly the ensemble techniques and deep learning components, requires substantial computational resources. This limitation might hinder its application in resource-constrained environments. Future work should focus on optimizing the model for efficiency and scalability. Furthermore, the complexity of the DMDI model might pose challenges in terms of interpretability and implementation. Simplifying the model without compromising its performance could be a direction for future research to make it more accessible to practitioners.

Enhancing the model's robustness against adversarial attacks on missing data imputation remains an open research area. Developing techniques to detect and mitigate such attacks will be crucial for deploying the model in high-stakes environments. Investigating the applicability of the DMDI model in other domains with missing data issues, such as healthcare or finance, could provide valuable insights and broaden the impact of this research. Implementing and testing the DMDI model in real-time IDS environments will be essential to evaluate its practical viability and effectiveness in operational settings.

The discussion emphasizes the broader implications of the DMDI model, acknowledges its limitations, and outlines future research directions. By providing a comprehensive analysis, this section helps readers understand the significance of our work in the context of ongoing developments in the field of IDS and missing data imputation.

5. Conclusion

This study emphasizes the crucial significance of missing data imputation specifically in the context of Intrusion Detection Systems (IDS) when the ratio of missingness in datasets is high. Deep_Learning_based_MissingData_Imputation (DMDI), efficiently handles missing data from the datasets thus increasing the input quality of machine learning models using a multi-stage imputation technique. Moreover, the Random Missing Values (RMV) algorithm provides a method to randomly introduce missing values into datasets, serving as a valuable tool for assessing and contrasting different approaches to managing absent data. We carried out a study on five distinct machine learning models to assess the impact of imputation. Our methodology disclosed varying results, with Support Vector Machines (SVM) showing a significant performance improvement. This highlights the considerable benefits of effective imputation techniques for models that rely on distance and hyperplane calculations. However, not all models encountered equivalent benefits, suggesting that achieving a universal method for managing missing values might not be attainable. Our research underscores the significance of meticulously choosing and tailoring an imputation technique that aligns with the distinctive characteristics of the dataset and the machine learning models employed.

Declaration of competing interest

There is no conflict of interest in this paper.

CRediT authorship contribution statement

Mahjabeen Tahir: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Azizol Abdullah:** Resources, Supervision, Validation, Writing – review & editing. **Nur Izura Udzir:** Supervision, Validation, Writing – review & editing. **Khairul Azhar Kasmiran:** Supervision, Validation, Writing – review & editing.

Acknowledgements

I am thankful to my co-authors for useful discussions.

Data availability

The associated data for this paper 10.57760/sciencedb.16599 can be accessed in the Science data bank database. (https://www.scidb.cn/en/s/3aEJbu).

Funding

Thank you to KeAi organization (Chinese Roots Global Impact) for supporting funding for researchers.

References

[1] O. Faker, E. Dogdu, (CICIDS2017 & Random Forest (RF) and Gradient Boosted Tree (GBT)) Intrusion detection using big data and deep learning techniques, in: *ACMSE 2019 - Proc. 2019 ACM Southeast Conf*, 2019, pp. 86–93.

[2] S. Shah, S. Pramod Bendale, An intuitive study: intrusion detection systems and anomalies, how AI can be used as a tool to enable the majority, in 5G era, in: *Proc. - 2019 5th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2019*, 2019, doi:10.1109/ICCUBEA47591.2019.9128786.

[3] S. Sanober, et al., An enhanced secure deep learning algorithm for fraud detection in wireless communication, Wirel. Commun. Mob. Comput. 2021 (2021), doi:10.1155/2021/6079582.

[4] S.A. Haque, M. Rahman, S.M. Aziz, Sensor anomaly detection in wireless sensor networks for healthcare, Sensors (Switzerland) 15 (4) (2015) 8764–8786, doi:10.3390/s150408764.

[5] R. Fujimaki, T. Yairi, K. Machida, An approach to spacecraft anomaly detection problem using Kernel Feature Space, in: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min, 2005, pp. 401–410, doi:10.1145/1081870.1081917.

[6] M. Ahmed, A.Naser Mahmood, J. Hu, A survey of network anomaly detection techniques, J. Netw. Comput. Appl. 60 (2016) 19–31, doi:10.1016/j.jnca.2015.11.016.

[7] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, Z. Xu, Missing value estimation for mixed-attribute data sets, IEEE Trans. Knowl. Data Eng. 23 (1) (2011) 110–121, doi:10.1109/TKDE.2010.99.

[8] L.E. Richards, R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data 26 (3) (1989).

[9] P. Lim, C.K. Goh, K.C. Tan, Evolutionary cluster-based synthetic oversampling ensemble (ECO-Ensemble) for imbalance learning, IEEE Trans. Cybern. 47 (9) (2017) 2850–2861, doi:10.1109/TCYB.2016.2579658.

[10] J. Yoon, W.R. Zame, M. Van Der Schaar, Estimating missing data in temporal data streams using multi-directional recurrent neural networks, IEEE Trans. Biomed. Eng. 66 (5) (2019) 1477–1490, doi:10.1109/TBME.2018.2874712.

[11] S. Zhang, Nearest neighbor selection for iteratively kNN imputation, J. Syst. Softw. 85 (11) (2012) 2541–2552, doi:10.1016/j.jss.2012.05.073.

[12] R. Pan, T. Yang, J. Cao, K. Lu, Z. Zhang, Missing data imputation by K nearest neighbours based on grey relational structure and mutual information, Appl. Intell. 43 (3) (2015) 614–632, doi:10.1007/s10489-015-0666-x.

[13] P.J. García-Laencina, J.L. Sancho-Gómez, A.R. Figueiras-Vidal, M. Verleysen, K nearest neighbours with mutual information for simultaneous classification and missing data imputation, Neurocomputing 72 (7–9) (2009) 1483–1493, doi:10.1016/j.neucom.2008.11.026.

[14] J. Josse, N. Prost, E. Scornet, and G. Varoquaux, "On the consistency of supervised learning with missing values," pp. 1–43, 2019, [Online]. Available: http://arxiv.org/abs/1902.06931.

[15] D.F. Swayne, A. Buja, Missing data in interactive high-dimensional data visualization, Comput. Stat. 13 (1) (1998) 15–26.

[16] S.G. Liao, et al., Missing value imputation in high-dimensional phenomic data: Imputable or not, and how? BMC Bioinformatics 15 (1) (2014) 1–12, doi:10.1186/s12859-014-0346-6.

[17] B.E.T.H. Twala, M.C. Jones, D.J. Hand, Good methods for coping with missing data in decision trees, Pattern Recognit. Lett. 29 (7) (2008) 950–956, doi:10.1016/j.patrec.2008.01.010.

[18] Y. Deng, T. Lumley, Multiple imputation through XGBoost, J. Comput. Graph. Stat. 0 (0) (2023) 1–19, doi:10.1080/10618600.2023.2252501.

[19] L. Gondara, K. Wang, MIDA: Multiple imputation using denoising autoencoders, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 10939 LNAI (2018) 260–272, doi:10.1007/978-3-319-93040-4_21.

[20] M.S. Santos, P.H. Abreu, S. Wilk, J. Santos, How distance metrics influence missing data imputation with k-nearest neighbours, Pattern Recognit. Lett. 136 (2020) 111–119, doi:10.1016/j.patrec.2020.05.032.

[21] S. van Buuren, K. Groothuis-Oudshoorn, mice: Multivariate imputation by chained equations in R, J. Stat. Softw. 45 (3) (2011) 1–67, doi:10.18637/jss.v045.i03.

[22] M. Tahir, A. Abdullah, N. Izura, K. Azhar, DeepImputeIDS: Enhancing intrusion detection systems with deep learning-based missing data imputation, in: 2023 14th Int. Conf. Inf. Commun. Technol. Syst. ICTS 2023, 2023, pp. 289–295, doi:10.1109/ICTS58770.2023.10330831.

[23] U. Garciarena, R. Santana, An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers, Expert Syst. Appl. 89 (2017) 52–65, doi:10.1016/j.eswa.2017.07.026.

[24] J. Yang, Y. Wang, Y. Yang, K. Ding, C. Na, Y. Yang, Effects of single and multiple imputation strategies on addressing over-fitting issues caused by imbalanced data from various scenarios, Appl. Intell. (2024) 9–11, doi:10.1007/s10489-024-05295-3.

[25] B. Halder, M.M. Ahmed, T. Amagasa, N.A.M. Isa, R.H. Faisal, M.M. Rahman, Missing information in imbalanced data stream: fuzzy adaptive imputation approach, Appl. Intell. 52 (5) (2022) 5561–5583, doi:10.1007/s10489-021-02741-4.

[26] A.M. Andrew, An introduction to support vector machines and other kernel-based learning methods, Kybernetes 30 (1) (2001) 103–115, doi:10.1108/k.2001.30.1.103.6.

[27] L. Folguera, J. Zupan, D. Cicerone, J.F. Magallanes, Self-organizing maps for imputation of missing data in incomplete data matrices, Chemom. Intell. Lab. Syst. 143 (2015) 146–151, doi:10.1016/j.chemolab.2015.03.002.

[28] L.P. Brás, J.C. Menezes, Improving cluster-based missing value estimation of DNA microarray data, Biomol. Eng. 24 (2) (2007) 273–282, doi:10.1016/j.bioeng.2007.04.003.

[29] D.T. Dinh, V.N. Huynh, S. Sriboonchitta, Clustering mixed numerical and categorical data with missing values, Inf. Sci. (Ny). 571 (2021) 418–442, doi:10.1016/j.ins.2021.04.076.

[30] W.C. Lin, C.F. Tsai, Missing value imputation: a review and analysis of the literature (2006–2017), Artif. Intell. Rev. 53 (2) (2020) 1487–1509, doi:10.1007/s10462-019-09709-4.

[31] M.H. Shahriar, N.I. Haque, M.A. Rahman, M. Alonso, G-IDS: Generative adversarial networks assisted intrusion detection system, in: *Proc. - 2020 IEEE 44th Annu. Comput. Software, Appl. Conf. COMPSAC 2020*, 2020, pp. 376–385, doi:10.1109/COMPSAC48688.2020.0-218.

[32] T.N. Dao, H. Lee, Stacked autoencoder-based probabilistic feature extraction for on-device network intrusion detection, IEEE Inter. Things J 9 (16) (2022) 14438–14451, doi:10.1109/JIOT.2021.3078292.

[33] H. Zhang, C.Q. Wu, S. Gao, Z. Wang, Y. Xu, Y. Liu, An effective deep learning based scheme for network intrusion detection, in: *Proc. - Int. Conf. Pattern Recognit*, 2018-Augus, 2018, pp. 682–687, doi:10.1109/ICPR.2018.8546162.

[34] C. Wang, C.T. Butts, J.R. Hipp, R. Jose, C.M. Lakon, Multiple imputation for missing edge data: a predictive evaluation method with application to Add Health, Soc. Networks 45 (2016) 89–98, doi:10.1016/j.socnet.2015.12.003.

[35] S.D.D. Anton, S. Sinha, H. Dieter Schotten, Anomaly-based intrusion detection in industrial data with SVM and random forests, in: 2019 27th Int. Conf. Software, Telecommun. Comput. Networks, SoftCOM 2019, 2019, pp. 1–6, doi:10.23919/SOFTCOM.2019.8903672.

[36] J. Gu, S. Lu, An effective intrusion detection approach using SVM with naïve Bayes feature embedding, Comput. Secur. 103 (2021) 102158, doi:10.1016/j.cose.2020.102158.
[37] C. Atik, R.A. Kut, R. Yilmaz, D. Birant, Support vector machine chains with a novel tournament voting, Electron 12 (11) (2023) 1–16, doi:10.3390/electronics12112485.
[38] M.A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke, Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study, J. Inf. Secur. Appl. 50 (2020) 102419, doi:10.1016/j.jisa.2019.102419.
[39] P. Dai, J. Luo, K. Zhao, H. Xing, X. Wu, Stacked denoising autoencoder for missing traffic data reconstruction via mobile edge computing, Neural Comput. Appl. 35 (19) (2023) 14259–14274, doi:10.1007/s00521-023-08475-3.
[40] N. Abedzadeh, M. Jacobs, and A. Definition, "A survey in techniques for imbalanced intrusion detection system datasets," vol. 17, no. 1, pp. 9–18, 2023.
[41] S. Choudhary, N. Kesswani, Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT, Procedia Comput. Sci. 167 (2019) (2020) 1561–1573, doi:10.1016/j.procs.2020.03.367.
[42] S. Bhatia, A. Jain, P. Li, R. Kumar, B. Hooi, MStream: Fast anomaly detection in multi-aspect streams, in: *Web Conf. 2021 - Proc. World Wide Web Conf. WWW 2021*, 2, 2021, pp. 3371–3382, doi:10.1145/3442381.3450023.
[43] V. Ukani, Parkinson's Disease Data Set, Kaggle (2020) [Online]. Available: https://www.kaggle.com/datasets/vikasukani/parkinsons-disease-data-set .
[44] UNBISCX NSL-KDD Dataset 2009, Canadian Institute of Cyber Secuirty, 2009 [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.
[45] UNSW, "The UNSW-NB15 Dataset," 2015. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset.
[46] A.D. Woods, D. Gerasimova, B. Van Dusen, J. Nissen, S. Bainter, A. Uzdavines, ... M.M. Elsherif, Best practices for addressing missing data through multiple imputation, Infant and Child Develop. 33 (1) (2024) e2407.
[47] G. Chhabra, V. Vashisht, J. Ranjan, A comparison of multiple imputation methods for data with missing values, Indian J. Sci. Technol. (2017).
[48] M. Templ, A. Kowarik, P. Filzmoser, Iterative stepwise regression imputation using standard and robust methods, Comput. Statist. Data Anal. 55 (10) (2011) 2793–2806.