Journal of Information Technology and Cyber Security 3(1) January 2025: 1-13 EISSN 2987-386X | PISSN 2987-3878
DOI: 10.30996/jitcs.12212


# Comparison of Dimensionality Reduction Techniques to Improve Performance and Efficiency of Logistic Regression in Network Anomaly Detection

Mokhamad Isna Marzuki Ahfa [1,*], Lukman Hakim [2], and Muhammad Imron Rosadi [3]

[1,2,3] Department of Informatics Engineering, Universitas Yudharta Pasuruan, Indonesia
* Corresponding author: 9isnaisna9@gmail.com

Received: 23 November 2024
Accepted: 10 January 2025
Revised: 16 December 2024
Available online: 14 January 2025

**To cite this article:** Ahfa, M. I. M., Hakim, L., & Rosadi, M. I. (2025). Comparison of Dimensionality Reduction Techniques to Improve Performance and Efficiency of Logistic Regression in Network Anomaly Detection. *Journal of Information Technology and Cyber Security, 3*(1), 1-13. https://doi.org/10.30996/jitcs.12212

## Abstract

Network anomaly detection is a crucial process to identify abnormal network traffic, which may pose a security threat. This research aims to improve the performance and efficiency of Logistic Regression (LR) in network anomaly detection by applying dimension reduction techniques, such as Principal Component Analysis (PCA), Truncated Singular Value Decomposition (TSVD), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Independent Component Analysis (ICA). The performance of each dimension reduction method is evaluated based on accuracy, precision, recall, F1-score, and computation time. The results show that TSVD provides the best performance with 95.86% accuracy, 0.96 precision, 0.96 recall, 0.95 F1-score, and 13.83 seconds computation time. In contrast, ICA showed the worst performance, especially in precision, recall, and F1-score, with values of 0.73, 0.83, and 0.78, respectively. Meanwhile, although t-SNE produces competitive accuracy, it has a high computational cost with an execution time of 1698.54 seconds. These findings show that choosing the right dimension reduction algorithm not only improves detection performance but also supports data processing efficiency, making it highly relevant for large-scale network security scenarios.

**Keywords:** dimensionality reduction, Logistic Regression, network anamoly detection, performance evaluation, Truncated Singular Value Decomposition.

## 1. Introduction

Nowadays, networking has become an essential component of human life. Almost every individual uses electronic devices, both wired and wireless, to utilize the internet network for various needs. Along with the increasing dependence on networks, network security aspects become very crucial to protect stored data from unauthorized access by unauthorized parties (Fikri & Djuniadi, 2021). Network security is not only to protect data, but also to ensure the smooth and reliable operation of systems that depend on the network.

In addition, in a network, there are conditions that can cause network traffic to be abnormal, known as network anomalies. Identification of these anomalies is critical to maintaining network integrity and security. Therefore, there is a need for an Intrusion Detection System (IDS) implemented on computer systems to detect and address potential threats that can damage the system (Imam, Sukarno, & Nugroho, 2019).

Anomaly detection is the process of finding patterns in a dataset that do not behave normally or do not match the desired expectations. This process is very important because it can identify data that is considered abnormal in a dataset. Anomaly detection has been widely studied in statistics and machine learning, with wide applications in various fields. Some examples of such fields include healthcare, fraud detection, intrusion detection, industrial defects, image processing, sensor networks, robot behavior, and astronomical data (Kwon, et al., 2019).

However, one of the main challenges in anomaly detection approaches is the high computational

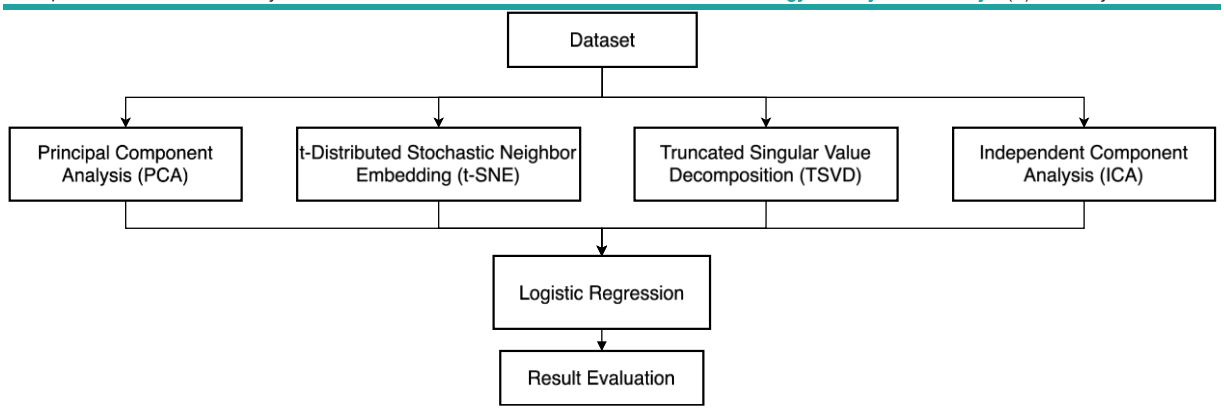


**Fig. 1.** Research flow.

complexity, especially when the dataset has a large number of features (high dimensionality), which can affect the performance of anomaly detection models. In addition, the more features in a dataset, the greater the risk that irrelevant information (noise) can dominate the data, which in turn can reduce the accuracy of the model. Therefore, this research highlights the importance of using dimensionality reduction strategies to handle such challenges. In this research, an anomaly detection technique with Logistic Regression algorithm and dimensionality reduction algorithm is used to improve the detection performance and accuracy (Gunawan, Sugiarto, & Mardianto, 2020).

Some of the algorithms used in this research include Principal Component Analysis (PCA), Truncated Singular Value Decomposition (TSVD), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Independent Component Analysis (ICA). PCA was chosen for its ability to identify the principal components that contribute most to the variation in the data, as well as its effectiveness on data with a linear structure (Hasan & Abdulazeez, 2021; Jolliffe & Cadima, 2016; Kurita, 2021). t-SNE is used for comparison because of its ability to visualize high-dimensional data and preserve local relationships between data, which makes it particularly suitable for complex non-linear data patterns (Silva & Melo-Pinto, 2023; van der Maaten & Hinton, 2008). ICA is also analyzed for its ability to decompose independent components in the data, which is relevant for detecting hidden patterns or anomalies that are not detected by linear methods (Hyvärinen & Oja, 2000; Jia, Sun, Lian, & Hou, 2022). Lastly, TSVD is also considered for its efficiency in handling large matrices and high sparsity data, while still maintaining a low-dimensional representation of the data (Golub & Van Loan, 2013; Tuo, Zhang, Huang, & Yang, 2021).

Previous research has shown the effectiveness of Logistic Regression in detecting anomalies in network data (Noureen, Bayne, Shaffer, Porschet, & Berman, 2019; Sasikala & Vasuhi, 2023). In the context of dimension reduction, PCA and t-SNE have also been shown to accelerate the processing of network anomaly data and improve the efficiency of classification algorithms (Hasan & Abdulazeez, 2021). In contrast to previous studies, this study focuses on the direct comparison of four dimensionality reduction algorithms, including TSVD, to identify the most efficient algorithm for network anomaly detection applications.

The hypothesis tested in this research is that the application of dimensionality reduction techniques will result in better anomaly detection performance (in terms of accuracy and computation time) compared to Logistic Regression without dimensionality reduction. This research provides new insights into the effect of dimensionality reduction algorithms on efficiency and accuracy in network anomaly detection, particularly on high-dimensional data often encountered in the context of network security. By comparing algorithms such as TSVD, PCA, and t-SNE, this study emphasizes the importance of selecting the optimal algorithm to improve data processing efficiency without compromising detection accuracy. The novelty of this study is its application to large network data that often requires fast processing, making it highly relevant for supporting real-time network security systems.

## 2. Methods

The framework shown in Fig. 1 illustrates the flow of research, starting from data collection, design, implementation of dimensionality reduction techniques, and finally, the implementation of network anomaly detection. Dimensionality reduction techniques used in this research include PCA, t-SNE, TSVD, and ICA algorithms. After the relevant features are successfully selected based on dimensionality reduction techniques, the next step is network anomaly detection using the Logistic Regression algorithm. The test

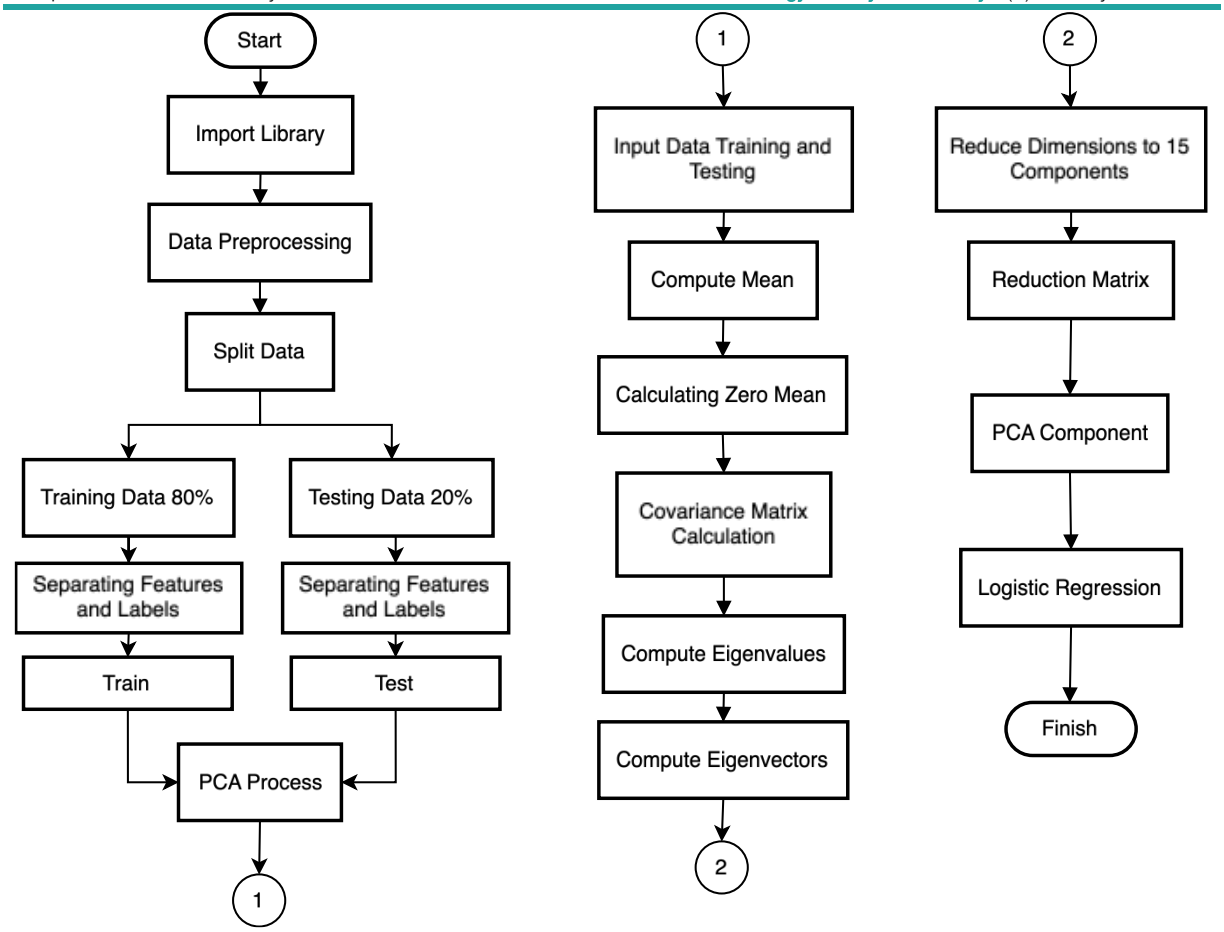


**Fig. 2.** Analysis in network anomaly detection using Logistic Regression.

results will then be evaluated to determine the most superior dimensionality reduction technique in detecting network anomalies using Logistic Regression.

## 2.1. Dataset

In this research, data collection was conducted through the literature study method. This method includes data acquisition from datasets as well as relevant journals that are the main references in the research (Erlin, Marlim, Junadhi, Suryati, & Agustina, 2022; Pramakrisna, Adhinata, & Tanjung, 2022; Utami, Nurlelah, & Hasan, 2021). This research uses the Python IDE on the Google Colab platform to monitor changes and activities that occur in networked systems. Various Python libraries are utilized, including Scikit-learn for machine learning modeling, pandas for data processing, and Matplotlib for visualization of test results.

Experiments were conducted using Network Anamoly Detection dataset from Kaggle (Onkarappa, 2019), which consists of 125,973 data with 43 attributes and covers 4 attack classes. This dataset includes various attributes related to network traffic information, such as duration, protocol, service, number of data bytes, and connection status (normal or error). The attack classes in this dataset are as follows:

1) Denial of Service (DoS): Attacks that drain the victim's resources, such as SYN flooding attacks.
2) Probing: Reconnaissance to gather information, such as port scanning.
3) U2R (User to Root): Unauthorized access to superuser, such as buffer overflow.
4) R2L (Remote to Local): Unauthorized access from remote machines, such as password guessing.

This dataset is divided into 80% train data and 20% test data. The initial data was processed using Logistic Regression algorithm without dimensionality reduction algorithm to see the initial performance. After that, processing is done with the dimensionality reduction algorithm combined with Logistic Regression to evaluate the improvement of model performance.

## 2.2. Principal Component Analysis (PCA)

PCA is the most commonly used linear dimension reduction algorithm. It transforms a high-dimensional dataset into a lower-dimensional form by projecting it towards principal components that optimally reflect the variance of the data. Thus, PCA can reduce the dimensionality of the dataset while re-

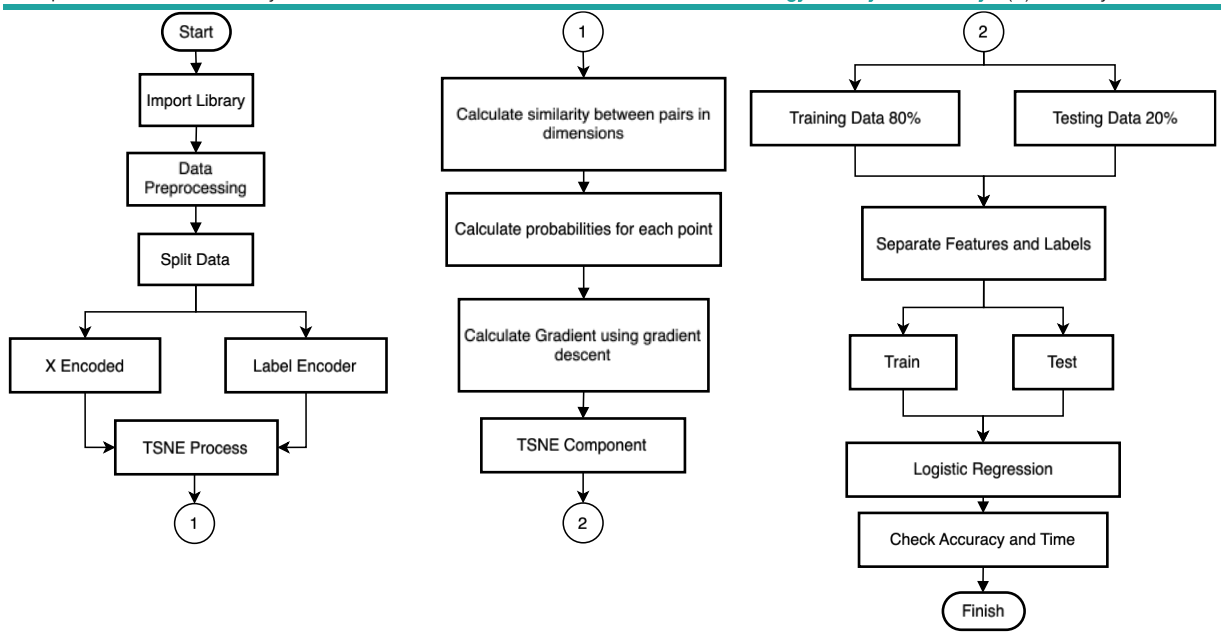


**Fig. 3.** Implementation of t-SNE for anomaly detection using Logistic Regression.

taining the key information in the data (Rhamadhani & Iswari, 2022). The application of PCA and Logistic Regression for network anomaly detection is presented in Fig. 2. The process begins with the Import Library step, which loads the Python libraries required for the analysis. After that, Data Preprocessing is performed to ensure the data is of sufficient quality to be used in modeling. The next step is to divide the dataset into two parts, namely Training Data (80%) which is used to train the model and Testing Data (20%) to evaluate the performance of the model. After splitting the dataset, the Separating Features and Labels step is performed, where attributes (features) are separated from targets (labels) to prepare the data for further processing by the applied algorithm.

In the main stage, PCA is used to reduce the dimensionality of the data to improve model efficiency and performance. The process starts with inputting training and testing data. After that, the average value of each feature is calculated (Calculating Mean) to determine the center of the data distribution. The next step is to convert the data into a zero-centered form (Calculating Zero Mean), so that the data distribution is more uniform and ready for further analysis.

The next step is to calculate the covariance matrix to understand the relationship between features (Covariance Matrix Calculation). Based on this matrix, Eigenvalue and Eigenvector calculations are performed, which yield important information about data variation. Principal components are then selected through the Reduce Dimension process to 15 Components, where the 15 most significant dimensions are stored in a Reduction Matrix. The result is data with smaller dimensions while still retaining important information, referred to as PCA Components. These PCA components become the input for the Logistic Regression algorithm, which is used to train the model to detect network anomalies. With the combination of PCA and Logistic Regression, the model is expected to identify anomalies efficiently and accurately.

### 2.3. t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a non-linear dimensionality reduction algorithm often used for high-dimensional data visualization. It maps data from a high-dimensional space to a low-dimensional space while maintaining the probability of similarity between points. For both linear and non-linear data, t-SNE is more effective than PCA in visualizing complex data structures (Devassy & George, 2020). The application of t-SNE and Logistic Regression for network anomaly detection is presented in Fig. 3. The process starts with importing the required libraries. Next, the data is processed through the Data Preprocessing stage to ensure that the data is in a ready-to-use format. After this stage, the data is divided into two subsets, namely Training Data (80%) to train the model and Testing Data (20%) to evaluate the model performance.

In the next stage, the t-SNE process begins with a similarity calculation between data pairs in the high dimension. The results of this calculation are then used to calculate the probability of each data point. After that, a gradient calculation is performed using the gradient descent method to optimize the position of data points in a lower dimensional space. This process aims to ensure that the relationship between data is well maintained after mapping. The end result of this process is the TSNE Component, which produces a

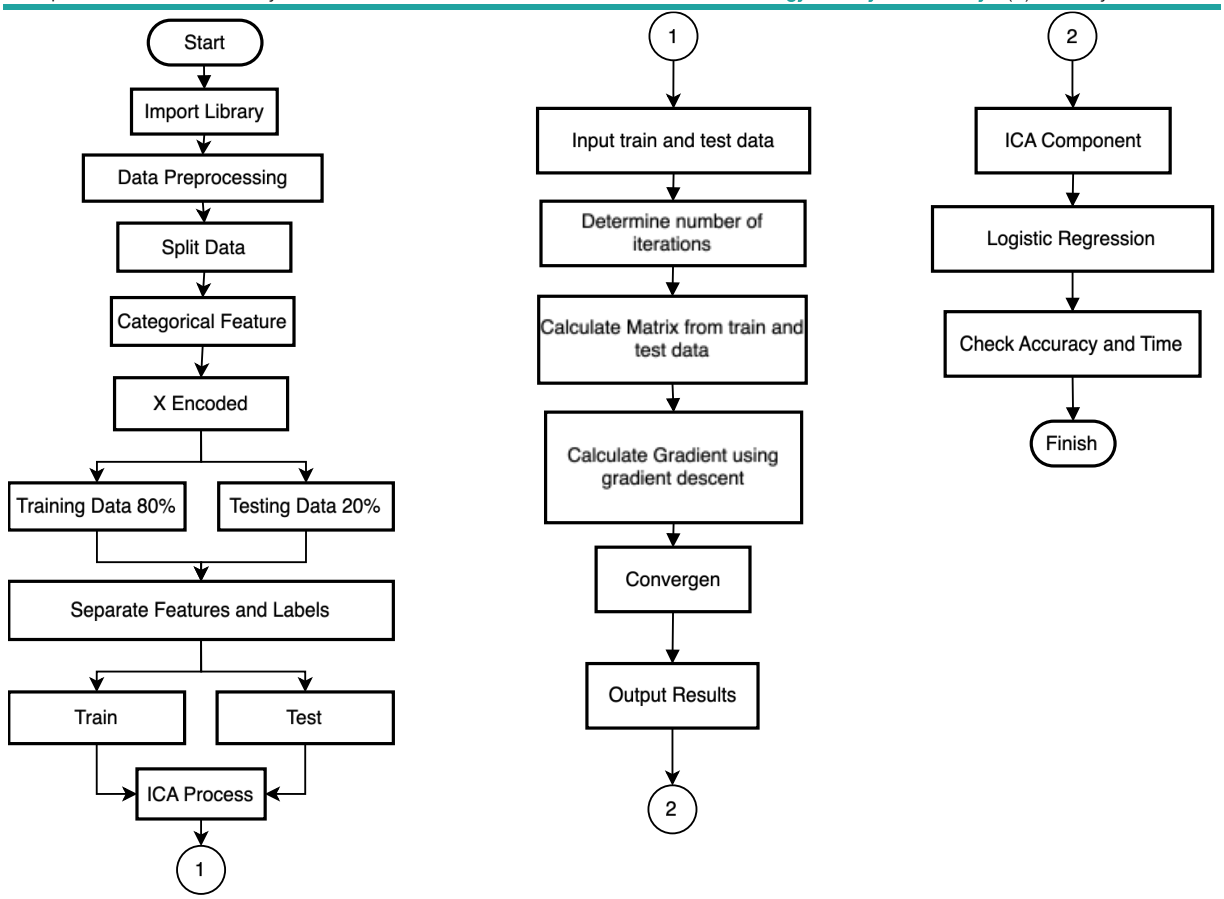


**Fig. 4.** Implementation of ICA for anomaly detection using Logistic Regression..

clearer and more structured mapping of the data in a low-dimensional space.

After the data is processed with t-SNE, the dataset is then subdivided into Training Data and Testing Data. At this stage, the features and labels for both subsets are separated. The model is then trained using the training data and tested using the testing data. After the training and testing stages, Logistic Regression is applied to predict the results based on the features that have been processed with t-SNE.

## 2.4. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a dimension reduction algorithm that focuses on separating independent signals from multi-dimensional data. This algorithm seeks to find combinations of independent signals that make up the data, which can help identify different sources of information in the dataset (Putra, Wiantari, Dewi, & Darmawan, 2019). The application of ICA and Logistic Regression for tissue anomaly detection is presented in Fig. 4.

The process starts with processing categorical features separately through the Categorical Feature Encoding stage, while the rest of the data is encoded in the X Encoded stage. After the encoding stage is complete, the dataset is divided into two subsets: training and testing data. These two subsets are then separated into features and labels required for model training and evaluation in the Train and Test stages.

## 2.5. Truncated Singular Value Decomposition (TSVD)

Truncated Singular Value Decomposition (TSVD) is a Singular Value Decomposition (SVD) based dimension reduction algorithm designed to handle data more efficiently. Unlike Principal Component Analysis (PCA), TSVD does not centralize the data before calculating the singular value decomposition, making it more suitable for sparse matrices that often appear in real-world applications (Akritidis & Bozanis, 2022). The process of applying TSVD and logistic regression for network anomaly detection is depicted in Fig. 5. It starts with data preparation through encoding, using either One Hot Encoder or Label Encoder, to convert categorical data into a numerical form that can be processed. Once the data is ready, the next step is the application of TSVD for dimension reduction. The TSVD process starts by calculating the SVD system matrix, which serves to simplify the data structure without losing important information. Through this decomposition, the data matrix is broken down into smaller components that are more manageable. TSVD then projects the data into a lower-dimensional space by solving the resulting system of linear equations.

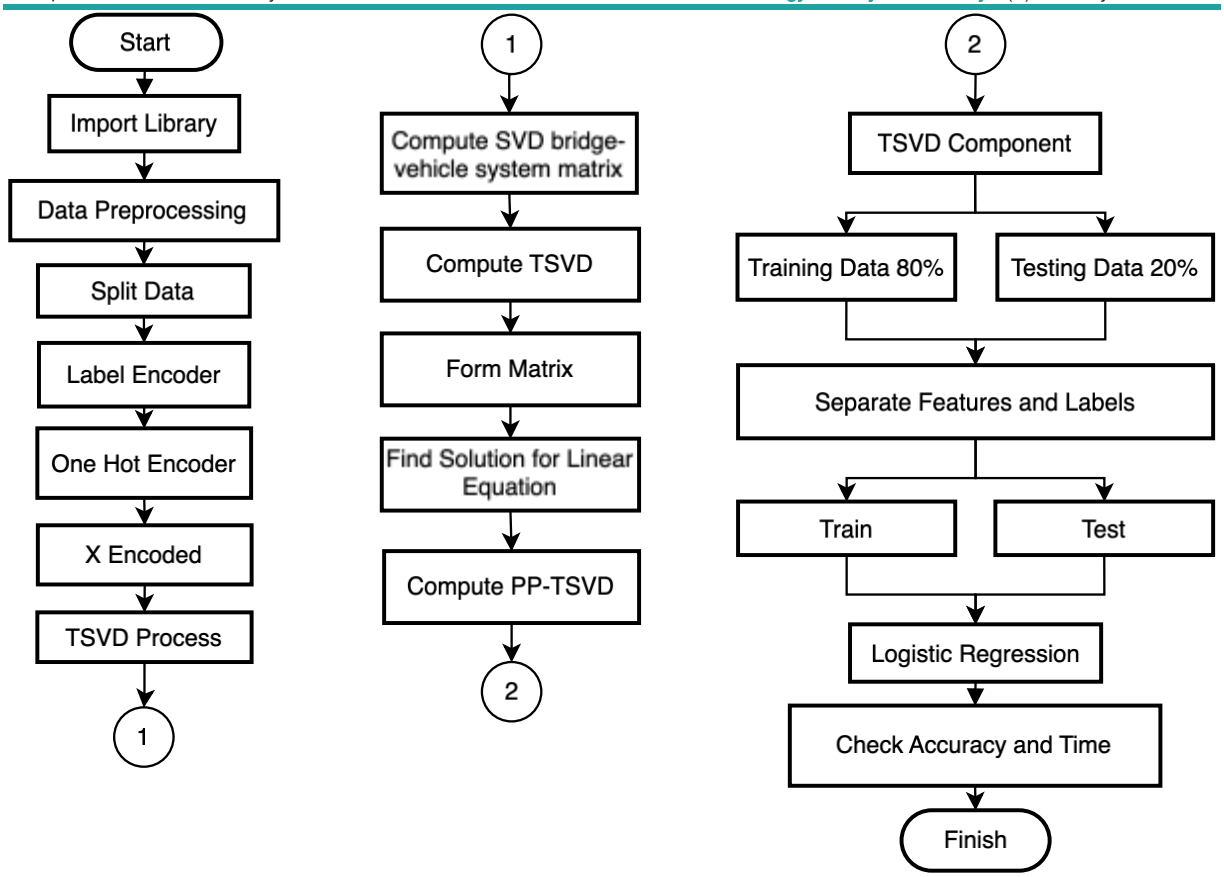


Fig. 5. Implementation of TSVD for anomaly detection using Logistic Regression.

This reduced dimensionality matrix retains key relevant information for further analysis while reducing the complexity of the data. The final result of the TSVD process is a reduction matrix that is used to train and test the logistic regression model. This step is followed by model evaluation through accuracy and inference time measurements, which aim to assess the overall efficiency and performance of the model. With this approach, TSVD not only helps manage high-dimensional data, but also supports the development of faster and more accurate models for network anomaly detection.

## 2.6. Logistic Regression

Logistic Regression is a simple yet powerful classification method. This method is often applied after the data has undergone a dimensionality reduction process using various algorithms (Willy, Rini, & Samsuryadi, 2021). The dimensionality reduction stage aims to simplify the data structure without losing important information, thus improving the efficiency of the classification process. The application of Logistic Regression to detect network anomalies is illustrated in the diagram in Fig. 6. In addition, the sequence of performance steps of the Logistic Regression algorithm is systematically described in Algorithm 1, which includes the process of data initialization, weight adjustment, and the final classification decision based on the resulting probabilities.

In Algorithm 1 (Kumar, 2021), where $i$ is the iteration of the logistic regression algorithm, starting from 1 to $k$. $k$ is the number of iterations performed in the training process. Typically, this process is run for the entire training data. $d_i$ is the $i$-th training data (instance). These are individual instances or samples in the dataset that are used to train the model. $y_j$ is the actual label or true class of the $j$-th data. The value of $j$ is usually 0 or 1 (for binary classification). $P(1|d_j)$ is the probability that the $j$-th data belongs to class 1, which is calculated by the logistic regression model. $z_i$ is the target for regression (Eq. 1). This value is the transformation of the difference between the actual labels $(y_j)$ and prediction probability $(P(1|d_j))$ which is divided by a probability-dependent factor. This variable helps direct the weight update. $w_j$ is the weight of the instance $d_j$, calculated based on the probability $P(1|d_j)$. These weights are used to give influence to certain data during training, especially if the probabilities are far from exact values. $f_j$ is a data-adjusted function based on the target value $(z_j)$ and weight $(w_j)$.

## 2.7. Evaluation of Results

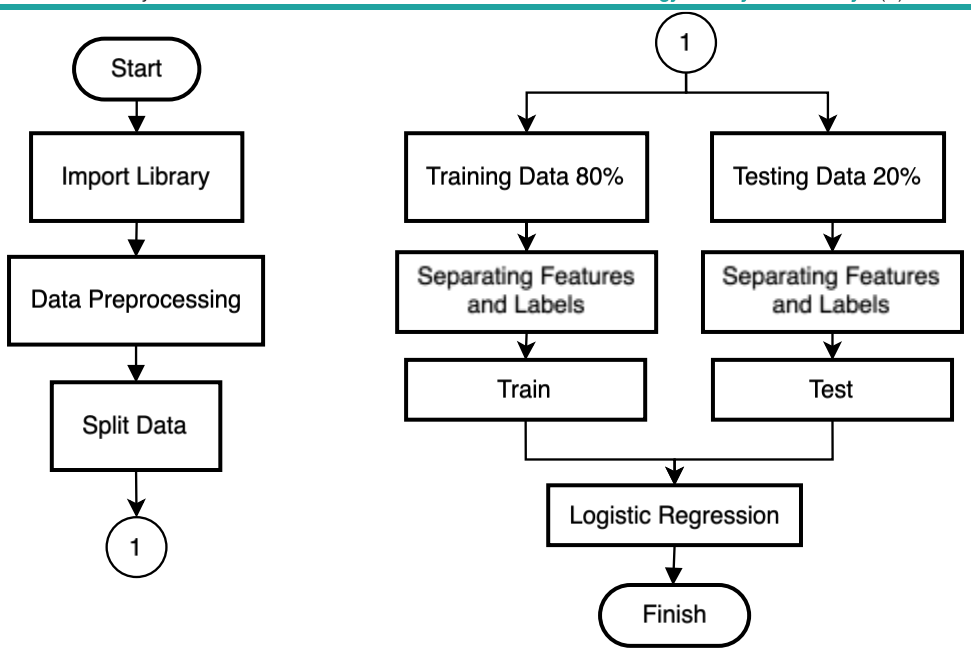


Fig. 6. Logistic Regression for Network Anomaly Detection.

**Algorithm 1.** Logistic Regression (Kumar, 2021).

1 Input: Training data
2 Begin
3 For $i = 1\ to\ k$:
4 For each training data instance $d_i$:
5 Set the target value for the regression to $z_i$ (Eq. 1)
6 Initialize the weight of the instance $d_j$ to $P(1|d_j).(1 - P(1|d_j))$
7 Finalize the function $f_j$ to fit the data with the class value $z_j$ and weight $w_j$
8 Assign class label 0 as Normal if $P(1|d_j) > 0.5$, otherwise assign class label 1 as Anomaly.

$$z_i = \frac{y_j - P(1|d_j)}{[P(1|d_j)(1-P(1|d_j))]} \quad (1)$$

The result of this detection is the level of accuracy produced by the Logistic Regression Algorithm method without dimension reduction and with dimension reduction. Comparison is made between performance based on confusion matrix and processing time using dimensionality reduction algorithms, namely PCA, t-SNE, ICA, and TSVD. The evaluation is based on accuracy, recall, precision, and f1-score values (Chicco & Jurman, 2020). The results of the classification model are obtained and presented in a Confusion Matrix. Confusion Matrix is used to show the comparison between the prediction results of the classification model and the actual data or label data. The calculation of accuracy refers to the equation formulated by Ruuska, et al. (2018). The Confusion Matrix scheme is presented in Fig. 7, which depicts the results of comparing the model predictions with the actual data in tabular form. This matrix consists of four main elements: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). True Positives (TP) refers to the case when the model correctly predicts the positive class, i.e. when the model successfully identifies objects or events that do belong to the positive category. False Positives (FP) occur when the model incorrectly predicts the positive class when the object actually belongs to the negative category. Conversely, True Negatives (TN) is when the model correctly predicts the negative class, i.e. when the model successfully identifies objects that should be classified as negative. False Negatives (FN) occur when the model incorrectly predicts the negative class when the object actually belongs to the positive category. These errors can pose a significant risk, especially in critical applications such as disease or security threat detection.

Accuracy is the percentage of the number of correctly predicted classes, which is the number of correct predictions for positive classes divided by the total number of predictions made. The accuracy value can be calculated using the formula presented in Eq. (2) (Ruuska, et al., 2018).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

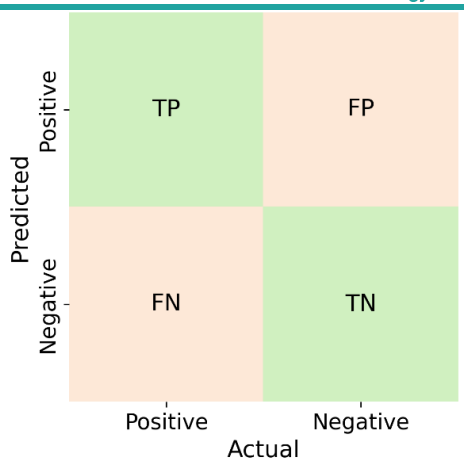


**Fig. 7.** Confusion Matrix (Zhang, Wang, An, Qin, & Yang, 2023).

Precision is the proportion of correct predictions for the positive class compared to the total predictions made for the positive class. The formula used to calculate the precision value is presented in Eq. (3) (Ruuska, et al., 2018).

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (3)$$

Recall or sensitivity, on the other hand, is the proportion or percentage of total positive class cases that are correctly predicted (Yacouby & Axman, 2020). This method is calculated by dividing the number of correct predictions for the positive class by the total number of actual positive class cases. The formula used to calculate recall is presented in Eq. (4) (Ruuska, et al., 2018).

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (4)$$

F1-score, which is the harmonic mean between precision and recall, is used to compare classifier performance, especially when there is a difference between precision and recall. The F1-score value can be calculated by the formula given in Eq. (5) (Gupta, Anjum, Gupta, & Katarya, 2021).

$$F1 - Score = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (5)$$

## 3. Results and Discussion

### 3.1. Results

After the experiment, the performance of the Logistic Regression algorithm without using dimensionality reduction shows an Area Under the Curve-Receiver Operating Characteristics (AUC-ROC) value of 80%, precision of 73%, recall of 87%, and F1-score of 79%. The resulting accuracy without dimensionality reduction reached 78.08% with a computation time of 197.75 seconds. Meanwhile, after applying dimension reduction techniques, namely PCA, t-SNE, ICA, and TSVD, the results obtained are as follows:

a. PCA
   PCA successfully reduced the number of features to 15 components, with an AUC-ROC score of 82%. The precision value was recorded at 73%, recall at 88%, and F1-score at 80%. The resulting accuracy reached 78.96%, with a computation time of 99.19 seconds. An example of the 15 components generated from PCA can be seen in Fig. 8.
b. ICA
   ICA successfully reduced the number of features to 10 independent components. An example of data resulting from feature reduction using ICA is presented in Fig. The precision value was recorded at 73%, recall at 83%, and F1-score at 78%. The resulting accuracy reached 82.89%, with a computation time of 60.98 seconds.
c. t-SNE
   t-SNE successfully reduced the number of features to 5 components, as shown in Fig.10. The precision value was recorded at 79%, recall at 84%, and F1-score at 81%. The resulting accuracy reached 84.60%, with a computation time of 1698.54 seconds.
d. TSVD
   TSVD successfully reduced the number of features to 10 components, as shown in Fig. The precision

| | PC1 | PC2 | PC3 | PC4 | PC5 |
|---|---|---|---|---|---|
| 0 | -1347.940940 | -2568.754453 | -29.780413 | 64.430497 | 40.407214 |
| 1 | -605.529225 | -2686.274001 | -46.189676 | -49.827530 | -165.771969 |
| 2 | 852.270406 | -2919.369602 | -31.932541 | 75.621850 | 34.451518 |
| 3 | -3098.214688 | -2289.547687 | -44.461009 | -63.111513 | -51.443140 |
| 4 | -3204.071735 | -2185.533095 | -29.418282 | -71.294808 | 79.571521 |
| ... | ... | ... | ... | ... | ... |

| | PC6 | PC7 | PC8 | PC9 | PC10 | PC11 |
|---|---|---|---|---|---|---|
| 0 | -65.987838 | 5.318424 | -26.808085 | 1.320379 | -1.711206 | -0.396201 |
| 1 | 90.803133 | -53.434993 | -3.045232 | 6.741201 | -2.436824 | 0.223900 |
| 2 | -69.830660 | 9.015878 | 13.081816 | -0.057160 | -1.473576 | -0.563581 |
| 3 | -84.167845 | 65.482592 | 14.218274 | 2.825593 | -0.323690 | 0.398147 |
| 4 | 7.411849 | 75.296661 | -17.394690 | -1.062965 | 1.502482 | 0.489796 |
| ... | ... | ... | ... | ... | ... | ... |

| | PC12 | PC13 | PC14 | PC15 |
|---|---|---|---|---|
| 0 | 0.114489 | -0.073416 | -0.390445 | -0.084108 |
| 1 | -0.634603 | 0.325926 | 0.550676 | -0.053380 |
| 2 | 0.094438 | -0.078573 | -0.110589 | -0.093493 |
| 3 | -0.350997 | -0.092069 | -0.245734 | 0.281610 |
| 4 | -0.327699 | 0.870259 | 0.266492 | 0.009024 |
| ... | ... | ... | ... | ... |

Fig. 8. Example of 15 components generated from PCA.

| | IC1 | IC2 | IC3 | IC4 | IC5 |
|---|---|---|---|---|---|
| 0 | 0.000042 | 0.000012 | 0.000303 | 0.000022 | -0.000621 |
| 1 | 0.000042 | 0.000016 | 0.000398 | 0.000012 | -0.000879 |
| 2 | 0.000046 | 0.000016 | 0.000219 | 0.000014 | -0.000466 |
| 3 | 0.000033 | 0.000017 | 0.000370 | 0.000011 | -0.001158 |
| 4 | 0.000025 | 0.000013 | 0.000149 | 0.000012 | 0.010273 |
| ... | ... | ... | ... | ... | ... |

| | IC6 | IC7 | IC8 | IC9 | IC10 |
|---|---|---|---|---|---|
| 0 | -0.000846 | 0.003680 | 0.000252 | 0.003818 | 0.000129 |
| 1 | -0.002900 | -0.001107 | 0.000161 | -0.003303 | -0.000384 |
| 2 | 0.000225 | 0.000133 | 0.000244 | -0.002300 | -0.006200 |
| 3 | -0.001232 | -0.003673 | 0.000241 | 0.004218 | 0.000101 |
| 4 | 0.002641 | -0.002095 | 0.000414 | 0.003576 | -0.001880 |
| ... | ... | ... | ... | ... | ... |

Fig. 9. Example of 10 components generated from ICA.

| | Dimension 1 | Dimension 2 | Attack |
|---|---|---|---|
| 0 | -64.601746 | 96.115311 | normal |
| 1 | -32.891300 | 79.554245 | normal |
| 2 | -29.484056 | -18.738459 | neptune |
| 3 | 86.424927 | 7.290921 | normal |
| 4 | 70.773827 | 70.484756 | normal |
| ... | ... | ... | ... |

Fig. 10. Example of 5 components generated from t-SNE.

| | Component 1 | Component 2 | Component 3 | Component 4 | Component 5 |
|---|---|---|---|---|---|
| 0 | 5.213991 | -0.501734 | 0.865255 | 0.825463 | 0.563054 |
| 1 | 5.112618 | 0.069002 | 1.068264 | 1.502929 | -0.203459 |
| 2 | 5.039790 | 2.854688 | -0.599854 | -0.014982 | 0.051518 |
| 3 | 4.733752 | -1.248471 | -1.300489 | -1.129602 | 0.286560 |
| 4 | 5.595271 | -1.615410 | -0.696240 | -0.610797 | -1.195324 |
| ... | ... | ... | ... | ... | ... |

| | Component 6 | Component 7 | Component 8 | Component 9 | Component 10 |
|---|---|---|---|---|---|
| 0 | -0.281981 | 0.053582 | 1.095909 | 1.209790 | 0.437046 |
| 1 | -0.327553 | 0.493919 | 0.195414 | -0.833879 | 0.845251 |
| 2 | 0.087416 | -0.012188 | 0.002041 | -0.059568 | 0.228579 |
| 3 | 0.294671 | 0.382525 | 0.511462 | -0.373783 | 0.112047 |
| 4 | 0.057836 | 0.114095 | -0.592103 | 0.200441 | 0.149649 |
| ... | ... | ... | ... | ... | ... |

Fig. 11. Example of 10 components generated from TSVD.

Table 1
Comparison of algorithm test results.

| Method | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | Computation Time (Seconds) |
|---|---|---|---|---|---|
| LR | 78.08 | 0.73 | 0.87 | 0.79 | 197.75 |
| PCA+LR | 78.96 | 0.73 | 0.88 | 0.80 | 99.19 |
| ICA+LR | 82.89 | 0.73 | 0.83 | 0.78 | 60.98 |
| t-SNE+LR | 84.60 | 0.79 | 0.84 | 0.81 | 1698.54 |
| TSVD+LR | **95.86** | **0.96** | **0.96** | **0.95** | **13.84** |

value was recorded at 96%, recall at 96%, and F1-score at 95%. The resulting accuracy reached 95.86%, with a computation time of 13.83 seconds.

### 3.2. Discussion

The use of dimensionality reduction, either before or after the application of the Logistic Regression algorithm, can have a significant impact on the results, depending on the use case and the data used. A comparison of the Logistic Regression algorithm results, both before and after using dimensionality reduction, is presented in Table 1.

In Table 2, the dimensionality reduction process using the t-SNE algorithm takes longer (1698.54 seconds) because it involves complex non-convex optimization to reduce the dimensionality of the data. t-SNE tends to take longer on large datasets due to such complexity. In contrast, the fastest dimensionality reduction algorithm is TSVD (13.83 seconds). This is because TSVD is a similar approach to PCA, but optimized for very large and sparse data matrices, as in the case of text data. With its optimized capability for sparse matrices, TSVD can be faster than PCA on text datasets with very high dimensions. This provides added value, especially when applied in real-world contexts.

The accuracy before and after using dimensionality reduction shows a significant improvement. However, each dimensionality reduction algorithm has different accuracy values. The highest value was achieved by TSVD with an accuracy of 95.86%. TSVD performs better than other methods because it is suitable for handling data with high dimensionality and many attributes that are not very important (sparse data). TSVD focuses on retaining the most influential parts of the data, so it can still summarize important information without losing much detail. Compared to other methods, such as PCA which relies more on the relationship between attributes, or t-SNE which is more suitable for visualization, TSVD proves to be more stable and efficient for use in anomaly detection models.

Prior to the application of dimensionality reduction, using all the original features (without dimensionality reduction) allows for a direct interpretation of the influence of each feature on the prediction results. This makes it easier to identify the features that contribute the most or least to the final result. However, if the dataset has many irrelevant or interrelated features (multicollinearity), the Logistic Regression algorithm may be prone to overfitting, which is when the model fits the training data too well but does not generalize well to new data. In addition, the more features in the dataset, the more complicated the calculations will be. This can lead to longer model training times and increased computational resource requirements.

After the dimensionality reduction process is performed, the number of features to be processed by the algorithm becomes fewer than the original, thus helping to reduce overfitting by retaining only the important features that contribute to the variability of the data. This improves the model's ability to generalize to new data. As a result, the training time of Logistic Regression algorithms tends to be faster. In addition, dimensionality reduction can help reduce noise and variance in the data, allowing the model to focus on more relevant patterns. However, it is important to remember that dimensionality reduction can also lead to loss of information from the original data, especially if done drastically. In addition, if the Logistic Regression algorithm is already good enough at handling high dimensions, applying dimensionality reduction may not always be necessary.

After being optimized with dimension reduction, the model can be deployed to a network monitoring system to analyze data traffic. In real scenarios, this application can be used to identify DoS, Probing, U2R, and R2L attacks by utilizing key features of the dataset, such as connection duration patterns, protocol types, and packet error rates. However, while dimensionality reduction algorithms such as TSVD help reduce complexity, the model still needs to be optimized to process data quickly without compromising detection accuracy.

## 4. Conclusions

Anomaly detection on the network using the Logistic Regression algorithm before the application of

dimensionality reduction resulted in an accuracy of 78.08% with a computation time of 99.19 seconds. After the addition of the TSVD dimension reduction algorithm, the best results were achieved with the highest accuracy value, which is 95.86%, and a faster processing time, which is 13.83 seconds. However, in this study there are still some limitations that need to be considered. This research is limited to binary classification (normal and anomaly), so it does not cover more specific attacks such as DoS, R2L, U2R, or Probing. In addition, the experiments were conducted entirely using Google Colab, which may affect performance and results if the same algorithms and techniques are applied in different hardware or software environments. This research also uses only one dataset from Kaggle, which has specific characteristics and may not represent various types of network anomaly data. The dataset has some weaknesses, such as data redundancy issues, an imbalance in the number of attacks, and a mismatch between the number of attacks and regular traffic. For future research, it is important to address the multiclass classification problem so that all dimension reduction algorithms can be tested more comprehensively. In addition, it is hoped that this data can be further developed for broader network security applications.

## 5. CRediT Authorship Contribution Statement

**Mokhamad Isna Marzuki Ahfa:** Data curation, Investigation, Methodology, Resources, Visualization, Writing–original draft, and Writing–review & editing. **Lukman Hakim:** Conceptualization and Methodology. **Muhammad Imron Rosadi:** Validation.

## 6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## 7. Data Availability

The dataset used in this study, titled Network Anamoly Detection, is publicly available on Kaggle and can be accessed at the following link: https://www.kaggle.com/datasets/anushonkar/network-anamoly-detection.

## 8. References

Akritidis, L., & Bozanis, P. (2022). How Dimensionality Reduction Affects Sentiment Analysis NLP Tasks: An Experimental Study. *Artificial Intelligence Applications and Innovations. 18*, pp. 301–312. Hersonissos, Crete, Greece: Springer. doi:https://doi.org/10.1007/978-3-031-08337-2_25

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics, 21*(6). doi:https://doi.org/10.1186/s12864-019-6413-7

Devassy, B. M., & George, S. (2020). Forensic Science International. *Forensic Science International, 311*. doi:https://doi.org/10.1016/j.forsciint.2020.110194

Erlin, E., Marlim, Y. N., Junadhi, J., Suryati, L., & Agustina, N. (2022). Early Detection of Diabetes Using Machine Learning with Logistic Regression Algorithm. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi, 11*(2), 88-96. doi:https://doi.org/10.22146/jnteti.v11i2.3586

Fikri, K. A., & Djuniadi, D. (2021). Keamanan Jaringan Menggunakan Switch Port Security. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan, 5*(2), 302-307. Retrieved from https://jurnal.uisu.ac.id/index.php/infotekjar/article/view/3501

Golub, G. H., & Van Loan, C. F. (2013). *Matrix Computations* (4th ed.). Baltimore, United States: Johns Hopkins University Press.

Gunawan, M. I., Sugiarto, D., & Mardianto, I. (2020). Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Seacrh pada Algoritma Logistic Regression. *JEPIN (Jurnal Edukasi dan Penelitian Informatika), 6*(3), 280-284. doi:https://doi.org/10.26418/jp.v6i3.40718

Gupta, A., Anjum, A., Gupta, S., & Katarya, R. (2021). InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray. *Applied Soft Computing, 99*. doi:https://doi.org/10.1016/j.asoc.2020.106859

Hasan, B. M., & Abdulazeez, A. M. (2021). A Review of Principal Component Analysis Algorithm for Dimensionality Reduction. *Journal of Soft Computing and Data Mining, 2*(1), 20-30. Retrieved from https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/8032

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks, 13*(4–5), 411-430. doi:https://doi.org/10.1016/S0893-6080(00)00026-5

Imam, R. M., Sukarno, P., & Nugroho, M. A. (2019). Deteksi Anomali Jaringan Menggunakan Hybrid Algorithm. *Proceedings of Engineering (E-Proceeding). 6*, pp. 8766-8787. Bandung, Indonesia: Universitas Telkom. Retrieved from https://core.ac.uk/download/pdf/299932449.pdf

Jia, W., Sun, M., Lian, J., & Hou, S. (2022). Feature dimensionality reduction: a review. *Complex & Intelligent Systems, 8*, 2663–2693. doi:https://doi.org/10.1007/s40747-021-00637-x

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374*(2065). doi:https://doi.org/10.1098/rsta.2015.0202

Kumar, V. (2021). Evaluation of computationally intelligent techniques for breast cancer diagnosis. *Neural Computing and Applications, 33*, 3195–3208. doi:https://doi.org/10.1007/s00521-020-05204-y

Kurita, T. (2021). *Principal Component Analysis (PCA).* Springer, Cham. doi:https://doi.org/10.1007/978-3-030-63416-2_649

Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., & Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. *Cluster Computing, 22*, 949–961. doi:https://doi.org/10.1007/s10586-017-1117-8

Noureen, S. S., Bayne, S. B., Shaffer, E., Porschet, D., & Berman, M. (2019). Anomaly Detection in Cyber-Physical System using Logistic Regression Analysis. *2019 IEEE Texas Power and Energy Conference (TPEC).* College Station, TX, USA: IEEE. doi:https://doi.org/10.1109/TPEC.2019.8662186

Onkarappa, A. (2019). Network Anamoly Detection. *Kaggle*. Retrieved from https://www.kaggle.com/datasets/anushonkar/network-anamoly-detection

Pramakrisna, F. D., Adhinata, F. D., & Tanjung, N. A. (2022). Aplikasi Klasifikasi SMS Berbasis Web Menggunakan Algoritma Logistic Regression. *Teknika, 11*(2), 90-97. doi:https://doi.org/10.34148/teknika.v11i2.466

Putra, A. P., Wiantari, N. W., Dewi, N. P., & Darmawan, I. D. (2019). Independent Component Analysis (ICA) dan Sparse Component Analysis (SCA) dalam Pemisahan Vokal dan Instrumen pada Seni Geguntangan. *JELIKU, 8*(1), 105-111. Retrieved from https://www.academia.edu/download/86504929/31504.pdf

Rhamadhani, M. H., & Iswari, L. (2022). Pengembangan Aplikasi Berbasis Web dengan R Shiny untuk Analisis Data Menggunakan Algoritma PCA. *Automata, 3*(1). Retrieved from https://journal.uii.ac.id/AUTOMATA/article/view/21870

Ruuska, S., Hämäläinen, W., Kajava, S., Mughal, M., Matilainen, P., & Mononen, J. (2018). Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. *Behavioural Processes, 148*, 56-62. doi:https://doi.org/10.1016/j.beproc.2018.01.004

Sasikala, K., & Vasuhi, S. (2023). Anomaly Based Intrusion Detection on IOT Devices using Logistic Regression. *2023 International Conference on Networking and Communications (ICNWC).* Chennai, India: IEEE. doi:https://doi.org/10.1109/ICNWC57852.2023.10127375

Silva, R., & Melo-Pinto, P. (2023). t-SNE: A study on reducing the dimensionality of hyperspectral data for the regression problem of estimating oenological parameters. *Artificial Intelligence in Agriculture, 7*, 58-68. doi:https://doi.org/10.1016/j.aiia.2023.02.003

Tuo, X., Zhang, Y., Huang, Y., & Yang, J. (2021). Fast Sparse-TSVD Super-Resolution Method of Real Aperture Radar Forward-Looking Imaging. *IEEE Transactions on Geoscience and Remote Sensing, 59*(8). doi:https://doi.org/10.1109/TGRS.2020.3027053

Utami, D. Y., Nurlelah, E., & Hasan, F. N. (2021). Comparison of Neural Network Algorithms, Naive Bayes and Logistic Regression to predict diabetes. *JITE (Journal of Informatics and Telecommunication Engineering), 5*(1), 53-64. doi:https://doi.org/10.31289/jite.v5i1.5201

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research, 9*(11), 2579-2605. Retrieved from https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

Willy, W., Rini, D. P., & Samsuryadi, S. (2021). Perbandingan Algoritma Random Forest Classifier, Support Vector Machine dan Logistic Regression Clasifier Pada Masalah High Dimension (Studi Kasus: Klasifikasi Fake News). *Jurnal Media Informatika Budidarma, 5*(4), 1720-1728. doi:https://doi.org/10.30865/mib.v5i4.3177

Yacouby, R., & Axman, D. (2020). Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models. *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP)* (pp. 79–91). Association for Computational Linguistics. doi:https://doi.org/10.18653/v1/2020.eval4nlp-1.9

Zhang, Z., Wang, W., An, A., Qin, Y., & Yang, F. (2023). A human activity recognition method using wearable sensors based on convtransformer model. *Evolving Systems, 14*, 939–955. doi:https://doi.org/10.1007/s12530-022-09480-y