IJHSR
ijhighschoolresearch.org
■ RESEARCH ARTICLE

# A Comprehensive Study on Anomaly Detection Methods Using Traditional and Machine Learning Approaches

Aditya Raj, Sanskar Sharma

Orion CBSE Senior Secondary School, Jalgaon, Maharashtra, 425306, INDIA; rajthakuraditya63@gmail.com

ABSTRACT: Detecting anomalies is critical in multiple domains such as cyber security, network security, etc. This paper provides an overview of anomaly detection methods, ranging from traditional statistical to modern ML approaches. The paper begins by explaining anomalies and their significance, followed by a detailed overview of traditional methods such as PCA and Z-Score Analysis. Then, there is a transition to methods based on machine learning that use supervised and unsupervised algorithms such as OCSVM & Bayesian Networks, followed by an illustration of the applications of these methods using examples in fraud detection. This paper also comparatively evaluates different methods, highlighting their advantages and limitations. In the end, this is a proposal for future research to address current limitations. This paper explains anomaly detection and its effective applications in various domains.

KEYWORDS: Robotics and Intelligent Machines; Machine Learning; Anomaly Detection; Cybersecurity.

## ■ Introduction

An anomaly, or outlier, is a data object that deviates significantly from most objects as if a different mechanism generated it.[1]

Anomaly detection is used in multiple fields, such as banking, healthcare, and the telecom industry, for various purposes, such as fraud detection. Traditional methods for anomaly detection, such as rule-based systems and clustering algorithms, have been used to identify anomalies. These methods often struggle to adapt to the complexities of modern data environments with diverse forms of anomalies. Moreover, the increase in the variety of data has posed significant challenges for these traditional approaches, leading to a greater risk of missed detections.

These days, researchers have turned to advanced technologies to enhance anomaly detection. Significant advancements have recently been witnessed in key areas like machine learning, which offers new approaches and advanced tools for detecting anomalies in high-dimensional data.

This paper discusses these technologies, their principles, and how they are used for anomaly detection. This paper focuses on machine-learning models, traditional methods, and architectures designed for anomaly detection in complex networks. By processing the insights from these technologies, this research paper contributes to anomaly detection and inspires research in this field in the future. (Section 1) contains details about past research done on anomaly detection. (Section 2) discusses how Anomaly Detection is used in real life. (Section 3) discusses some traditional approaches and methods for anomaly detection. Furthermore, (Section 4) discusses various machine-learning algorithms for anomaly detection. (Section 5) contains a comparison of multiple modern methods, stating their limitations and benefits. Experiments and results are shown in (Section 6).

### *1. Past research in the Anomaly Detection field.:*

**Table 1:** Past research in Anomaly Detection.

| S.No. | References | Models | Year | Work done |
|---|---|---|---|---|
| 1. | Brauckhoff D. *et al.*[17] | PCA | 2009 | Applying PCA for traffic anomaly detection. |
| 2. | Iglewicz B. *et al.*[19] | Z-Scored | 1993 | Modified Z-Scored method for enhancing outlier detection. |
| 3. | Hodge *et al.*[21] | IQR | 2004 | Emphasize the importance of traditional methods like IQR in providing a baseline in more complex models |
| 4. | Chandola *et al.*[22] | IQR | 2009 | Found that the robustness of IQR against outliers laid the groundwork for developing advanced algorithms. |
| 5. | Anne George *et al.*[25] | SVM | 2012 | To construct classifiers in IDSs and evaluate the performance of models using the KDD99 data set |
| 6. | Wenjie Hu *et al.*[26] | SVM | 2003 | A robust support vector machine was used to detect anomalies in noisy data. Reconciles the problem of overfitting due to noisy data sets used in training the mode |
| 7. | Yang Li *et al.*[28] | KNN | 2007 | They use TCM-KNN for pattern recognition, fraud detection and outlier detection. |
| 8. | Jing Tian *et al.*[27] | KNN | 2014 | The k-nearest neighbor algorithm was used to identify the observation of test data whose BMUs were removed to reduce the influence of noise. |
| 9. | Imran N. Junejo.[30] | BN | 2010 | Used BN in scene modeling for performing video surveillance |
| 10. | Linquan Xie *et al.*[31] | FCM | 2010 | Used FCM for detecting anomalies in the normal flow |
| 11. | Jose F. Nieves *et al.*[35] | K-Means | 2009 | They evaluated their work of maintaining a low false alarm rate that leads to a high detection rate. |
| 12. | B. Kiran Kumar *et al.*[36] | K-Means | 2012 | Used K-Means for scanning network flow data using different attributes like IP address, Protocol, Port Number etc. for detecting anomalies |
| 13. | Zhongfeng Wang *et al.*[39] | OCSVM | 2019 | OCSVM was used combined with the PSO algorithm to improve the convergence speed by adaptive speed weighting and adaptive population splitting. |

(Table 1) contains past research from their published year and what work they have done.

### *2. Practical applications of Anomaly detection:*

Anomaly detection is important in various applications ranging from fraud detection[2] to medical diagnosis.[3,4] Researchers have used modern machine-learning models to detect anomalies. This section contains several applications of anomaly detection.



DOI: 10.36838/v6i11.2

***2.1 Fraud detection:***

Commercial organizations like banks, credit card companies, and so on use fraud detection often powered by anomaly detection, to detect criminal activities. This is for crimes like identity theft, where a malicious user can be an actual customer of the organization or someone posing to be an actual customer. Organizations are interested in quick action against such frauds to prevent economic losses.

***2.1.1 Credit Card Fraud Detection:***

This domain uses anomaly detection methods to detect fraudulent credit card usage. This is similar to the detection of insurance fraud.[5]

In this case, the data consists of records such as User ID, consecutive card usage in a specific time frame, amount spent, etc. Fraud is typically shown in transactional records. Specifically, purchases of items never purchased by the customer, high purchase rates, and so on.[6]

***2.1.2 Insider Trading Detection:***

This is specifically a part of the stock market, where people make illegal profits by leaking inside information before it is made public. This information could include anything that would affect the stock prices in a particular industry. The available data is from various sources. This domain uses anomaly detection methods to detect fraud online to prevent organizations from making illegal profits.[6] Traditional methods, in this case, can miss minor deviations in the data and not detect the anomalies.

***2.2 Cybersecurity:***

Today's networks are increasingly susceptible to intrusions and attacks because of various access points, with anomaly detection playing a crucial role in identifying these threats. These days, even disgruntled employees and even terrorist factions are getting involved in cybercrimes. Network-based attacks are rising as software and protocols become increasingly vulnerable and attacks continue to advance.[9]

***2.2.1 Intrusion detection:***

Nowadays, Political and commercial entities are seen to be getting into cyber-warfare to damage or censor the information on computer networks.[7] Intrusion detection is the process that helps monitor the events occurring in a computer system or network and analyze them for signs of possible incidents; it often prohibits unauthorized access.[8]

Traditional intrusion detection systems depend on predefined known attack patterns to identify threats. These systems cannot detect novel or previously unseen attacks, which may generate false outcomes or miss subtle deviations from already-known attack patterns.

***3. Traditional methods for Anomaly Detection:***

Traditional anomaly detection methods include statistical techniques like Principal Component Analysis,[10] These methods have been used in various applications, including network traffic monitoring, hyperspectral imagery, and time-series data analysis. Recently, Jia proposed a method combining supervised learning with statistical analysis for anomaly detection in time-series data.[11] These traditional methods provide a foundation for developing more advanced anomaly detection methods.

***3.1 Statistical Methods:***

Multiple anomaly detection methods have been introduced in recent research. Ray proposed IoTDixon, a scheme that uses Dixon's Q test to analyze small-scale data and detect point anomalies.[12] Zhang proposes a method for inspecting industry product quality based on the Gaussian Restricted Boltzmann Machine.[13] Deepthi provided an overview of these methods, including Grubb's, Dixon's Q Test & the Z-Score Method.[14] Principal Component Analysis has mainly been effective in the domain of anomaly detection, as demonstrated by Issariyapat.[15]

***3.1.1 Principal Component Analysis:***

PCA is a method used for anomaly detection with the help of an intrusion predictive model that uses both major and minor principal components of data. Anomalies detection is based on their distance from normal instances in the principal component space, identifying outliers that deviate significantly from the expected patterns. This approach has been said to achieve high precision and recall in identifying anomalies, outperforming several other detection methods.[16]

***3.1.2 Z-Score Method:***

It is the statistical method that measures the standard deviation of a data point from its mean. First, this method calculates the difference of data points from the mean of data sets and then divides it by the standard deviation of the database. This gives us a Z-score value. Then, define a threshold; points beyond this threshold are considered anomalies. The most common value of the threshold is ±3, Which indicates that if the data point value is greater than plus or minus 3, it's an outlier.[18]

$$Z = \frac{x - \mu}{\sigma}$$

*Eq.1*

Eq.1) is the formula for calculating the z-score, where 'x' is the data points, '$\mu$' is the mean, and 'σ' is the standard deviation.

Iglewicz B. and Hoaglin D. introduced the modified Z-score (mZ-score) method for robust outlier detection. They focused on improving the classical Z-score by utilizing robust scale estimators, such as the Median Absolute Deviation (MAD) and Sn. The modified Z-score method, leveraging these estimators, enhances outlier detection by reducing sensitivity to extreme values that can skew traditional measures like the mean and standard deviation.[19]

***3.2 Interquartile Range (IQR) Method:***

The IQR Method uses the spread of the middle 50 % of the data to detect anomalies. First, calculate the first quartile(Q1) and then the third quartile (Q3). Q1, the first quartile, is the median of the first half of the data set, and Q3, the third quartile, is the median of the third half of the data set. Then, calculate the IQR.

$$IQR = Q3 - Q1$$

Eq.2

Define the Lower Bound and Upper Bound. Data Points outside these boundaries are considered anomalies. It mea

sures the variability that is less sensitive to outliers compared to standard deviation.[20]

$$Lower\ Bound = Q1 - 1.5\,IQR$$

$$Upper\ Bound = Q3 + 1.5\,IQR$$

*Eq.3*

(Eq.2) is the formula for calculating the IQR, where 'Q3' and 'Q1' are the first and third quartiles, respectively. (Eq.3) is the formula for calculating Lower and Upper Bound.

Hodge and Austin emphasize the importance of traditional methods like IQR in providing a baseline in more complex models. They argue that models like IQR are still relevant because of their simplicity and effectiveness in data analysis.[21]

Chandola, Banerjee, and Kumar went deeper into the evolution of anomaly detection methods and described the foundational role of IQR in early detection systems. They found that the robustness of IQR against outliers laid the groundwork for developing advanced machine-learning algorithms.[22]

***4. The Preferred approach to anomaly detection using Machine Learning:***

Machine learning (ML) is a field of study in artificial intelligence concerned with developing and studying statistical algorithms that can learn from data and generalize to unseen data, thus performing tasks without explicit instructions. Unlike traditional statistical methods, which often rely on predefined models and assumptions about data distribution, ML techniques can adapt to complex, high-dimensional datasets. This flexibility makes ML particularly effective in anomaly detection tasks, such as cybersecurity and intrusion detection.

***4.1 Supervised Anomaly Detection:***

For constructing a predictive model, supervised methods, i.e., classification methods, are trained on labeled data sets containing normal and abnormal samples. Although the detection rate of these models is better compared to semi-supervised and unsupervised methods because of the access to more information, it contains some issues that make this method less accurate than they are supposed to be. The first issue is the shortage of data sets on which the model is trained. Additionally, the biggest challenge is to obtain accurate labels. Also, training sets contain some noise, which leads to higher false alarm rates.[23] Different supervised algorithms are Regression, Supervised Neural Networks, Support Vector Machines, K-nearest Neighbors, Bayesian Networks, and Decision Tree.

***4.1.1 Support Vector Machines (SVM):***

Vapnik proposed the Support vector machines (SVM).[24] It is a model that uses classification algorithms for two-group classification and regression analysis. It is known for producing great accuracy on less computational power. In SVM, the support vector stands for a data point or node lying closest to the decision boundary or hyperplane, thus playing a vital role in defining the decision boundary and the margin of separation. Moreover, SVM provides a technique called Regularization to prevent overfitting. It introduces a penalty term in the objective function to encourage finding a more straightforward decision boundary rather than fitting the training data perfectly.

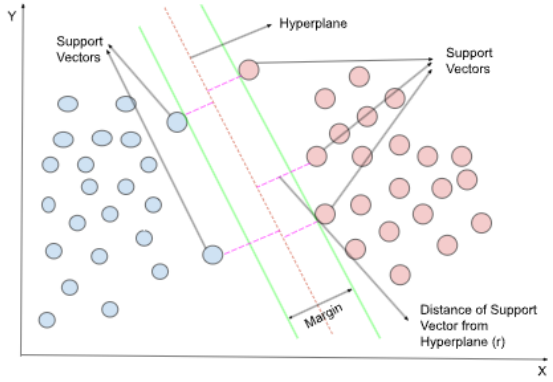


**Figure 1:** GENERAL Graph of Support Vector Machine (SVM). By looking at the figure, we can see how SVM classifies data.

(Figure 1) shows the general graph of SVM. This figure clearly visualizes the hyperplane, support vectors, and margin.

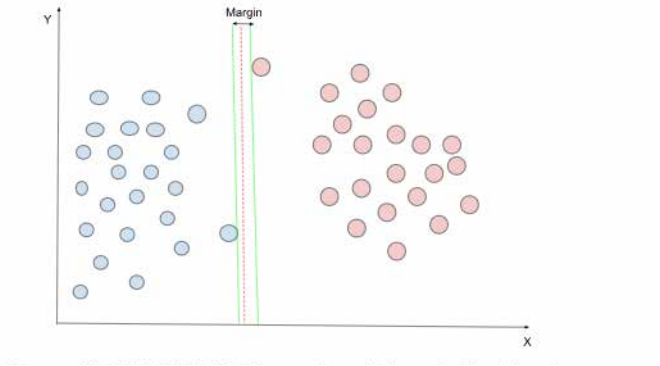


**Figure 2:** DIFFERENT Hyperplane taken in the Graph. By seeing this figure, we can find out that if the hyperplane is taken at 90°; the SVM cannot classify the data correctly.

(Figure 2) shows the classification done by SVM in a different way as compared to (Figure 1). In this figure, the Hyperplane is taken at an angle of 90°.

To choose the correct hyperplane. See the margin value. The greatest value is the correct hyperplane. (Figure 1) Margin > (Figure 2) Margin. Hence, (Figure 1) is the correct gra*ph.*

$$f(x) = sign(w^t.x + b)$$

*Eq.4*

$$w^t.x + b = -1$$

*Eq.5*

$$w^t.x + b = 1$$

*Eq.6*

$$w^t.x + b = 0$$

*Eq.7*


DOI: 10.36838/v6i11.2

$$r = \frac{\left|w^t.x_i + b\right|}{\|w\|}$$

Eq.8

(Eq.4) is the equation of the decision function of SVM where 'f(x)' represents the output of the decision function for input vector x. 'sign()' gives +1 when the argument is positive and -1 when the argument is negative and 0 if the argument is zero. It is used to determine the class label. *' w tx'* is the dot product of the weight vector w and the input vector x. It represents the projection of x onto the hyperplane defined by w. It determines which side the hyperplane lies on. 'b' is the bias term. It represents the offset of the hyperplane from the origin along the direction of the normal vector w and also shifts the hyperplane away from the origin.

(Eq.5), (Eq.6), and (Eq.7) represent the margins of a linear support vector machine. '$w^t x$' is the dot product of the weight vector w, and the input vector x. 'b' represents a bias term or intercept. It shifts the decision boundary away from the origin along the direction defined by w. The constant value -1 or +1 describes whether the margin is left or right, and 0 represents the decision boundary or hyperplane.

(Eq.8) is the formula to calculate the distance of a support vector from the hyperplane.$w^t$ represents the transpose of the weight vector w. '$x_i$' is the feature vector of the support vector. 'b' stands for the bias term. $|w^t. xi+ b|$ gives the absolute value and $|(|w|)|$ represents the Euclidean norm of w.

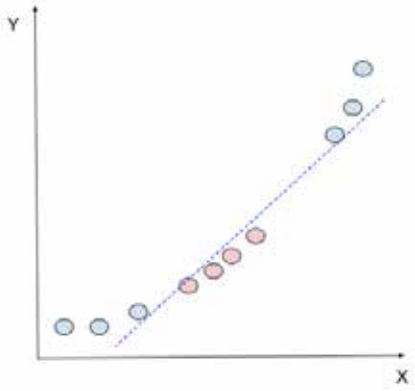


**Figure 3:** GRAPH of Non-Linear Support Vector Machine. By looking at this figure, we can see that the decision boundary effectively divides two classes, demonstrating SVM's high accuracy in nonlinear data.

(Figure 3) displays how SVM handles non-linear problems. In (Figure 3) we can clearly see how SVM has classified like and unlike terms above and below hyperplane, respectively.

Annie George used multi-class Support Vector Machines to construct classifiers in IDSs and evaluated the performance of models using the KDD99 data set. They use multi-class to construct multiple two-cause SVM classifiers and then combine their classification results with the voting method.[25]

Hu W, Liao Y., and Vemuri used a Robust Support Vector machine to detect noisy data anomalies. It reconciles the problem of overfitting due to noisy data sets used in training the model. Additionally, it is fast, makes the decision surface smooth using standard support vector machines, and automatically controls the amount of regularisation.[26]

***4.1.2 K-Nearest Neighbors (K-NN):***

K- nearest neighbor is a nonparametric supervised machine learning algorithm used to identify the nearest neighbors. Then, the observation in the training data close to the nearest neighbor is classified into a class using the classification method.[27] The main idea is that similar data points tend to have similar labels or values. While creating this classifier, (k), also known as a hyperparameter, is a significant parameter; change in values of k leads to a difference in performance. As the value of k increases, the classification time increases, affecting prediction accuracy.[23]

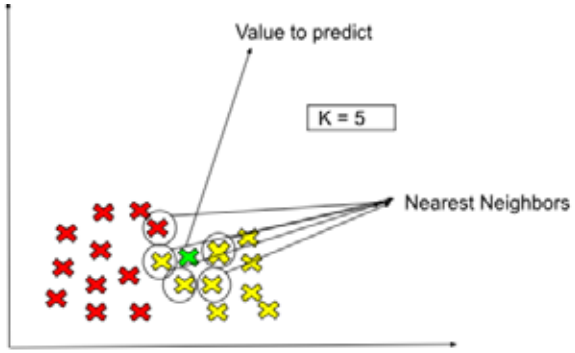


**Figure 4:** GRAPH of KNN. By seeing this figure, we can find out how nearest neighbors are formed to predict the value for classification.

In (Figure 4), hyperparameter = 5, which means our model will take five nearest data sets around the value that will be predicted, i.e., Nearest Neighbors. Then, the distance of the test value from each neighbor is calculated. The distance can be calculated using two methods: Euclidean and Manhattan. The Test value gets the value of the closest neighbor as the predicted value.

Li, Y., Fang, B., Guo, L., and Chen, Y. used a novel anomaly detection method based on the TCM-KNN (Transductive Confidence Machines for K-Nearest Neighbors) algorithm. They use it for pattern recognition, fraud detection, and outlier detection.[28]

Tian, J., Azarian, M.H. and Pecht, M. used the k-nearest neighbor algorithm to identify the observation of test data whose BMUs are removed to reduce the influence of noise; then they calculated the Euclidean distance between the test data observation and the centroid of the neighbors as an anomaly indicator.[27]

***4.1.3 Bayesian Networks (BN):***

A Bayesian network is a probabilistic graphical model representing a set of variables and their conditional dependencies using a directed acyclic graph. They are used to model the relationships between different variables in a data set, thus identifying the unusual patterns or behaviors that deviate from the expected norm. BN allows anyone to understand causal relationships and handle incomplete data sets. Additionally, BN provides an efficient method for avoiding overfitting of data as compared to other models.[29]

Imran N. Junejo used BN in scene modeling to perform video surveillance. During training, the BN extracted the dis

criminative path from the identified paths that are used by most objects in the scene, and then these networks classify the incoming path by taking reference from their size, location, speed, acceleration, and spatial-temporal curvature characteristics of other objects while testing them.[30]

***4.1.4 Regression:***

It is estimating the relationship between dependent and one or more independent variables. If more than one independent variable is taken, it's known as multiple linear regression. However, it predicts the value of the dependent variable based on the values of the independent variable; one can also use it for classification by using logistic regression. There are various types of regression: linear regression, Polynomial regression, logistic regression, etc. The most common among them is Linear regression. In Regression analysis, one can measure the difference between the predicted and actual values by seeing the Root Mean Square Error value. RMSE helps to identify whether the data is overfitting or not.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}\right)^2}$$

Eq.9

(Eq.9) is the formula to calculate the Root Mean Squared Error in regression analysis. 'n' represents the total number of observations in your data set. '$y_i$' represents the observed (actual) value of the dependent variable for the i-th observation '$\hat{y}_i$' represents the predicted value of the dependent variable for the i-th observation. '$\Sigma$' denotes the summation operator, which means you will sum up the values for all observations from i=1 to n. '$1/n$' is the reciprocal of the total number of observations representing the average.

If the data is linear, meaning its value is linearly increasing, then linear regression is used. Otherwise, polynomial regression is considered the best choice for non-linear data. However, it takes time to train on the data set. The R-squared method is used to calculate the accuracy of the model. The value of R-squared is between 0 and 1. 0 is the worst value, and 1 is the best value.

$$R^2 = 1 - \frac{\sum_i \left(y_i - \hat{y}_i\right)^2}{\sum_i \left(y_i - \bar{y}\right)^2}$$

*Eq.10*

(Eq.10) is the formula for calculating the coefficient of determination($R^2$). 'n' represents the total number of observations in your data set. '$y_i$' represents the observed (actual) value of the dependent variable for the i-th observation. '$\hat{y}_i$' represents the predicted value of the dependent variable for the i-th observation. '$\Sigma$' denotes the summation operator, which means you will sum up the values for all observations from i=1 to n. '$1/n$' is the reciprocal of the total number of observations representing the average. '$\underline{y}$' represents the mean of the dependent variable's observed (actual) values.

$$y = \beta_0 + \beta_1 x$$

*Eq.11*

(Eq.11) represents the linear regression equation where 'y' is the dependent variable, which will predict, and 'x' is the dependent variable, a feature used to predict values. '$\beta_0$' represents the y-intercept of the regression line, representing the value of y when x is 0. '$\beta_1$' is the slope of the regression line.

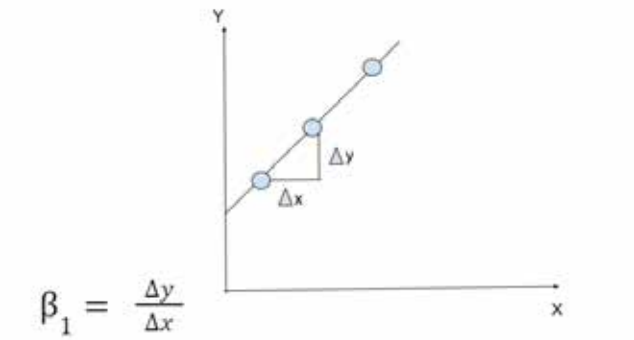


**Figure 5:** GRAPH of Linear Regression. This figure shows that to find any predicted value, a best-fit line is calculated by calculating the slopes of a parameter.

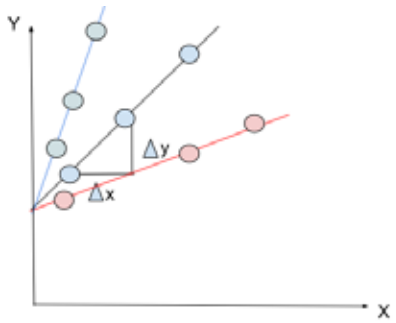


**Figure 6:** GRAPH of Multiple Regression. By seeing this figure, we can find out that if we have to predict multiple parameters, then each parameter's best-fit line is calculated.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n + \varepsilon$$

*Eq.12*

Figure 5) shows a general graph of linear regression displaying the slope and best-fit line. (Figure 6) shows the graph of multiple regression, displaying multiple parameters and best-fit line.

(Eq.12) represents the multiple linear regression equation, where 'y' is the target or dependent variable. '$x_1$, $x_2$, $x_3$, . . ., $x_n$' are independent variables or features. '$\beta_0$' represents the intercept term, and '$\beta_1$, $\beta_2$, $\beta_3$, . . ., $\beta_n$' represents the coefficients of independent variables. 'ε' represents the error term.


DOI: 10.36838/v6i11.2

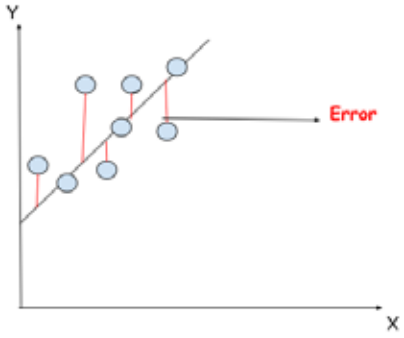


**Figure 7:** ERROR in the graph of regression. By looking at this figure, we can see how errors are calculated based on the distance of points from the best-fit line.

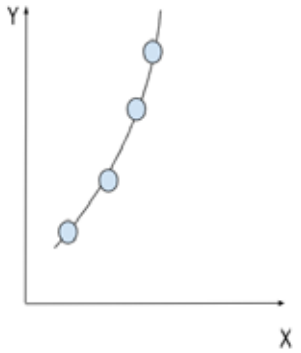


**Figure 8:** GRAPH of Polynomial Regression. By seeing this figure, we can see that this model effectively identifies the increasing trend and suggests a positive polynomial relationship.

(Figure 8) shows the graph of polynomial regression. From the graph, we can infer that it is used when the relation between the independent and dependent variables is not linear.

$$y = \beta_0 + \beta_1 x_1^1 + \beta_2 x_2^2$$

*Eq.13*

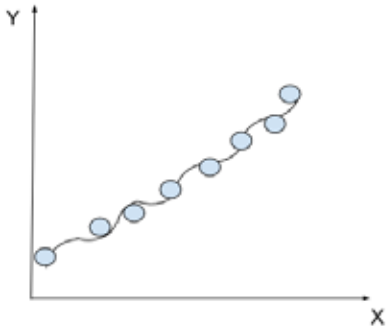


**Figure 9:** GRAPH of Polynomial Regression up to n terms. By looking at this figure, we can clearly see that adding more polynomial terms doesn't decrease accuracy. The model fits the data significantly, thus showing that it can handle complex relationships.

(Figure 9) shows the graph of polynomial regression up to n terms. If the relation between independent and dependent variables is cubic, bi-quadratic, etc., we use polynomial regression up to n terms.

$$y = \beta_0 + \beta_1 x_1^1 + \beta_2 x_2^2 + \beta_3 x_3^3 + \ldots + \beta_n x_n^n + \varepsilon$$

*Eq.14*

(Eq.13) and (Eq.14) represent the equation of polynomial regression, where y and x represent the dependent and independent variables, respectively. '$\beta_0$' represents the intercept term, and '$\beta_1, \beta_2, \beta_3, \ldots, \beta_n$' represents the coefficients of independent variables. 'ε' represents the error term. 'n' represents the polynomial degree.

**Table 2:** The EXAMPLE data set of Classification.

| Student Study Hours | Pass or Fail |
|---|---|
| 4 | Pass |
| 2 | Fail |
| 1 | Fail |
| 5 | Pass |
| 0.5 | Fail |

Table 2) has a sample data set in which students' results are given based on their study hours. The following is the classification of whether the student has passed or failed through linear and logistic regression.

As linear regression only predicts values, it cannot be used in classification, so we will label Pass as 1 and Fail as 0. If the number is less than 0.5 and greater than 0, the student fails; if it is between 0.5 and 1, the student is Passed.

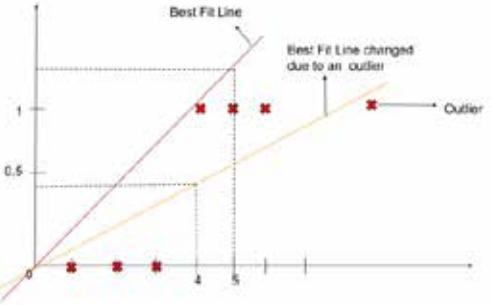


**Figure 10:** LIMITATION of Linear regression. This figure explains the limitations of linear regression and why we use logistic regression for classification. Therefore, using linear regression can lead to Value error.

Limitation of using linear regression:

Outliers: It changes the Best fit line, which leads to wrong predictions. As shown in (Figure 10) even after studying for 4 hours, the value less than 0.5 means the student has failed.

Value Error: The best-fit value can even predict negative values and values greater than 1, which makes no sense.

(Figure 10) represents the limitations graphically. Therefore, linear regression can't be used for classification problems and hence cannot be used for anomaly detection. Consequently, using Logistic Regression in anomaly detection is the best choice.

$$h_0(x) = \frac{1}{1 + e^{-\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n\right)}}$$

*Eq.15*

(Eq. 15) represents the equation of logistic regression, where '$\beta_0$' represents the intercept term and '$\beta_1, \beta_2, \beta_3, \ldots, \beta_n$' represents the coefficients of independent variables. 'e' is the base of the natural logarithm.

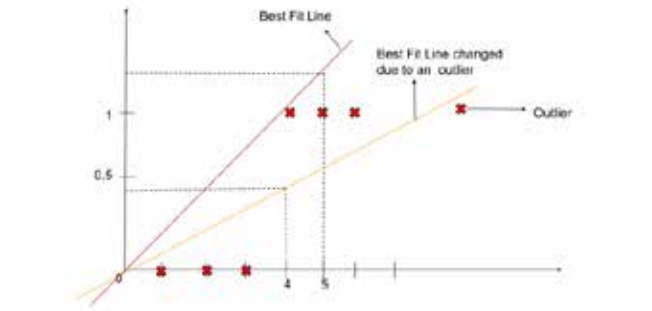


**Figure 11:** GRAPH of Logistic Regression. The figure clearly shows that logistic regression is able to solve the limitations of linear regression in classification problems. It is stashed from the top and bottom, which helps the model restrict the domain of predicted value from 0 to 1.

### *4.2 Unsupervised Anomaly Detection:*

Unsupervised methods are trained on unlabeled data sets. It facilitates the analysis of raw data sets and helps generate analytic insights from them. Interest in unsupervised machine learning is leveraged because of the expensiveness of labeled data sets in supervised learning, as labeled data sets are generally unavailable, and the manual annotation will be inconvenient and a waste of time. The combination of unsupervised and supervised is called semi-supervised learning. The benefit of semi-supervised learning is that it can preprocess the data before analysis, resulting in building a good feature representation. Additionally, it helps find patterns and structures in unlabeled data.[31] Different unsupervised algorithms are Fuzzy C-Means (FCM), One-Class Support Vector Machines, K-Means, etc.

#### *4.2.1 Fuzzy C-Means (FCM):*

The fuzzy C-Means clustering algorithm is one of the most classic algorithms in unsupervised machine learning. The benefit of using this is that it does not need to mark the category information of data records in advance. On the contrary, it also has cons, like it depends extensively on the initial value and tends to fall into local minima. Additionally, it usually requires more iterations. Numerous iterations like FCM with genetic algorithm (GA) generate new optimal individuals by selecting the best individual and mutating the operation. This process continues till the optimal initial clustering center is generated; it has two benefits: first, it makes it easy to converge to the local minimum, and second, it improves its dependence on high value.[32] Linquan Xie, Ying Wang, Liping Chen, and Guangxue Yue used FCM to detect anomalies in normal flow. They used various hybrid methods with FCM to optimize it. Some of them are average information entropy, support vector machines, etc.[33]

#### *4.2.2 K-Means:*

It is a clustering model that groups objects based on the values of their features into clusters. It's very important to define the number of clusters. After that, divide all the objects into K clusters, compute their centroids, and verify that each centroid differs. This should be done to initialize the K clusters centroids. Next, calculate the distance between the centroids and all clusters, and then assign each object to the cluster with the closest centroid. Finally, the centroid of both mutated clusters will be recalculated until the value stops being changed.[34] Steps are displayed in (Figure 12).

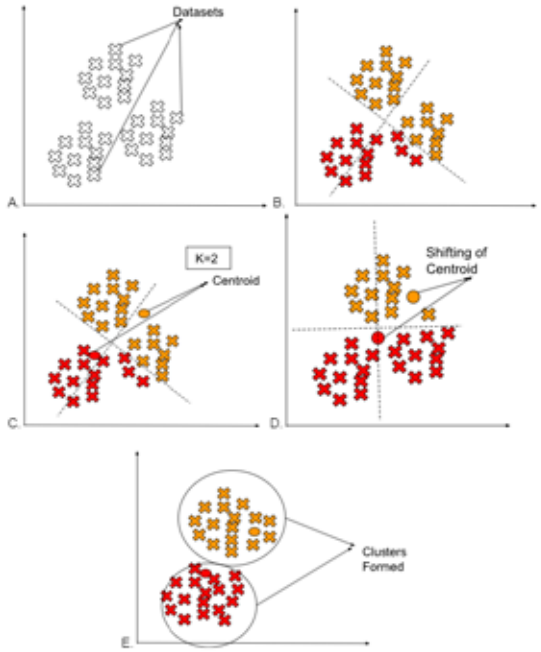


**Figure 12:** VARIOUS Steps For KNN. The figure shows that to solve the classification problem, the model finds the centroid first, then shifts the centroid, and then clusters are formed for classification.

Jose F. Nieves used the K-means algorithm with KDD Cup 1999 to evaluate the performance of unsupervised learning methods for anomaly detection. The evaluation of this work was that maintaining a low false alarm rate leads to a high detection rate.[35]

To detect anomalies, B. K. Kumar and A. Bhaskar used K-Means to scan network flow data using different attributes like IP address, protocol, port number, etc. They used visualization to identify which sites are more frequently accessed by users.[36]

#### *4.2.3 One-Class Support Vector Machine (OCSVM):*

One-Class Support Vector Machines (OCSVM) is one of the most popular approaches for anomaly detection because of its flexibility and ability to find various nonlinear borders that divide data classes.[37] It maps nominal data to high-dimensional kernel space and divides it from the origin with maximum margin to find the boundaries for making decisions.[38]

Zhongfeng Wang, Yatong Fu, Chunhe Song, Peng Zeng, and Lin Qiao used OCSVM combined with the PSO algorithm to improve the convergence speed by adaptive speed weighting and adaptive population splitting. It supports the algorithm in jumping out of the local optimum and thus improves the classification accuracy of OCSVM.[39]


DOI: 10.36838/v6i11.2

### *5. Comparison between Machine Learning Models in Anomaly Detection.*

**Table 3:** COMPARISON between ML models. The table gives us a basic idea of the pros and cons of every ML model used for anomaly detection.

| S.No. | Algorithm | Benefits | Limitations |
|---|---|---|---|
| 1. | Logistic Regression | It can be performed on low-end computers. It's fast compared to other models | When the feature space is large it may give inaccurate results. |
| 2. | Support Vector Machine | It can deal with high-dimensional data. It works well in anomaly detection. | • To get more accurate, choosing the right kernel is a must; this can take time<br>• Takes a long time and large computational power.<br>• During constraint solving, some numerical stability problems can occur.<br>• Requires both +ve and -ve data. |
| 3. | K-Nearest Neighbors | • It is very easy to learn and implement.<br>• Can be useful in building models that contain non-numerical data types. For instance, Texts. | • Requires much time and computational costs.<br>• Poor performance was observed during the analysis of unbalanced samples. |
| 4. | Bayesian Networks | With the ability to recognize patterns in gas levels over time, one can lower the chances of false alarms triggered by fixed thresholds that fail to account for normal fluctuations. This advantage of Bayesian networks helps us better detect real anomalies. | It might sometimes miss important information or get things wrong, which can make their predictions and make them less reliable. |
| 5. | Fuzzy C-Means | • It lets a single data point belong to more than one group, giving a better picture of how genes behave.<br>• It shows how genes act more naturally by letting them be part of different groups at the same time. | Determine the membership cutoff value and the number of clusters(c). |
| 6 | K-Means | • It is easy and less complex.<br>• The K-means algorithm must always figure out the distances between every pair of data points. Importantly, it doesn't insist that these distances follow specific rules like being metric. | • Have to specify the value of k.<br>• Outliers can drastically affect the model. |
| 7. | OCSVM | It achieves a more accurate model than conventional methods. | It depends upon labeled data sets, which may not always be available. |

## ■ Results and Discussions

This section gives you a detailed view of the practical experiments conducted to evaluate the effectiveness of various anomaly detection methods. This paper employs credit card fraud detection through various modern and traditional methods.

### *6.1 Data Description:*

It only contains numerical data variables that result from a PCA transformation. Features V1, V2, and V28 are the principal components obtained with PCA; the 'Time' and 'Amount' have not transformed with PCA. The feature 'Amount' contains transaction amounts. Feature 'Class' tells whether the transaction is a fraud, 0 refers to a legitimate transaction, and 1 refers to a fraudulent transaction. There are 31 columns and 284808 rows in the dataset.

### *6.2 Experimental Setup:*

All experiments are conducted using Python. Various Python libraries like Scikit-learn, Matplotlib, Seaborn, Pandas, and Numpy have been used for this paper. Scikit-learn is used to implement machine learning; Matplotlib and Seaborn make graphs and confusion matrices, while numpy and pandas are used for data preprocessing. Each method's ability to detect anomalies in various data sets was evaluated.

### *6.3 Results and Analysis:*

#### *6.3.1 Performance Metrics:*

**Table 4:** Accuracy Percentage of Different Models. The table shows that KNN performs best in anomaly detection compared to all other models.

| Method | Type | Accuracy (%) |
|---|---|---|
| Logistic Regression | Modern | 93.61 |
| K-Nearest Neighbor (KNN) | Modern | 99.88 |
| K-Means | Modern | 95 |
| Fuzzy C-Means (FCM) | Modern | 99.7 |
| Support Vector Machine (SVM) | Modern | 99.36 |
| Interquartile Range (IQR) | Traditional | 53.88 |
| Z-Scored | Traditional | 88.79 |
| PCA | Traditional | 95.04 |

The accuracy of different methods is shown in (Table 4).

#### *6.3.2 Figures and Graphs:*

##### *6.3.2.1 Logistic Regression:*


**Figure 13:** Confusion Matrix of Logistic Regression. The confusion matrix demonstrates high true positive rates for correctly identifying legitimate classes. It also has minimal misclassification.

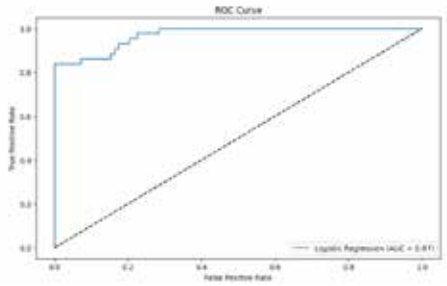


**Figure 14:** ROC curve of logistic regression. ROC curve has an excellent area under the curve of 0.97, indicating logistic regression has high capabilities to distinguish between the classes (legitimate and fraud).

***1. (Figure 13) Confusion Matrix:***

- **Purpose:** Shows the model prediction rate.
- **True Positives:** 36 Fraud transactions are correctly predicted.

• **True Negatives:** 96 Legitimate transactions are correctly predicted.

• **False Positives**: 2 Legit transactions wrongly predicted as fraud.

• **False Negatives:** 7 Fraud transactions wrongly predicted as legit.

***2. (Figure 14) ROC Curve:***

• **Purpose:** This graph evaluates the performance of the logistic regression model by plotting the true positive rate against the false positive rate at various threshold settings.

• **AUC (Area Under Curve):** This measures the model's overall performance. Here, the AUC is 0.97, indicating excellent accuracy. A perfect model has an accuracy of 1.0.

***6.3.2.2 K-Nearest Neighbor:***

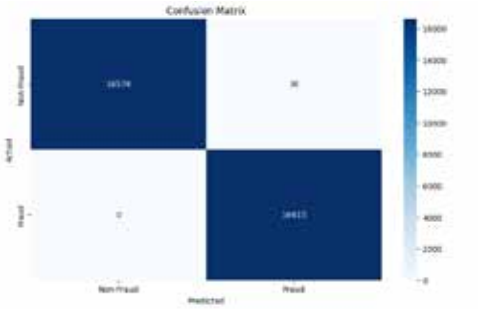


**Figure 15:** KNN Confusion Matrix. This figure clearly shows the KNN model's outstanding performance in Fraud Detection, with no false negatives and only 38 false positives, demonstrating its high precision in identifying Deepfakes.

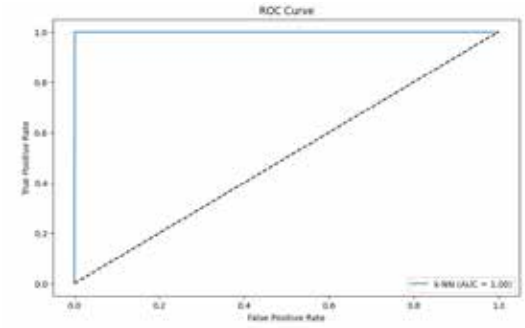


**Figure 16:** ROC curve of KNN. A perfect area under the curve (AUC) of 1.0 indicates that the KNN model perfectly distinguishes between fraudulent and legitimate cases without any error.

• **(Figure 16) ROC Curve:** The model has an AUC of 1.00, which indicates that it is close to being perfectly accurate.

• **(Figure 15) Confusion Matrix:** The model shows excellent predictive performance with negligible errors, making it extremely good for detecting fraudulent transactions.

***6.3.2.3 K-Means:***

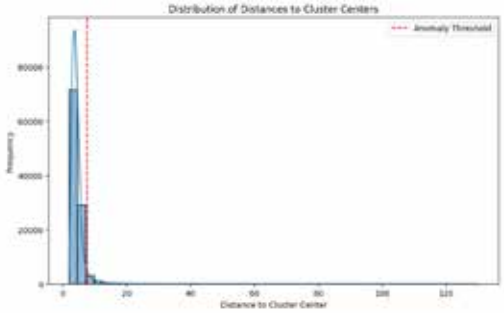


**Figure 17:** Distribution of Distances to Cluster Centers. This histogram shows the anomaly threshold (displayed as a dotted line), which indicates that most data points are tightly clustered around.

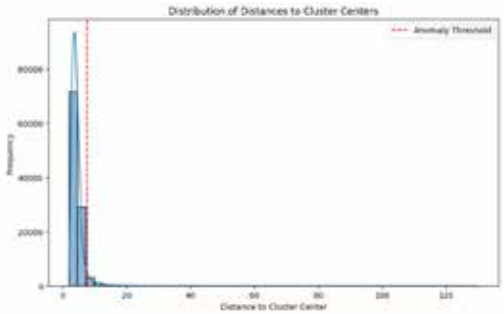


**Figure 18:** Cluster Visualization with PCA. This figure shows the spatial distribution that shows the K-Means algorithm has successfully identified and grouped data points into clusters based on their similarities.

• **(Figure 17) Distribution of Distances to Cluster Centers:** This histogram shows the distribution of data points to their cluster centers. From the graph, it can be concluded that most points have a low distance to their cluster center, indicating good clustering. Additionally, the red dashed lines represent the anomalies threshold, above which points are anomalies, and the long straight line on the right side of the distribution shows the outliers.

• **(Figure 18) Cluster Visualization with PCA:** The scatter plot shows the clusters formed by the k-means algorithm. The PCA is applied to reduce feature dimensions. The orange points represent anomalies.

**6.3.2.4 Fuzzy C-Means:**

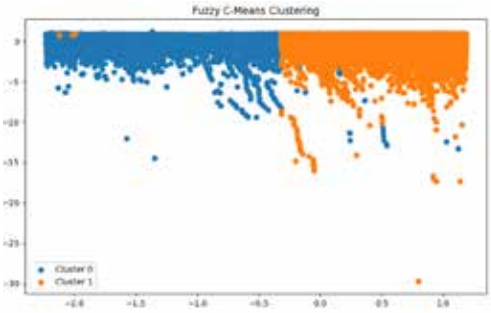


**Figure 19:** Cluster Visualization with Fuzzy C-Means. In this figure, we can see two overlapping clusters with gradient membership, indicating that data points belong to both clusters rather than a strict classification.

(Figure 19) The graph distinguishes between different transaction patterns. Each point is colored according to its highest membership grade, i.e., the cluster to which it mainly belongs. Points on the left side of the plot likely display anomalies.


DOI: 10.36838/v6i11.2

#### 6.3.2.5 Support Vector Machine:

**Figure 20:** SVM Confusion Matrix. This figure shows the excellent performance of SVM, which has only 45 false negatives and no false positives.

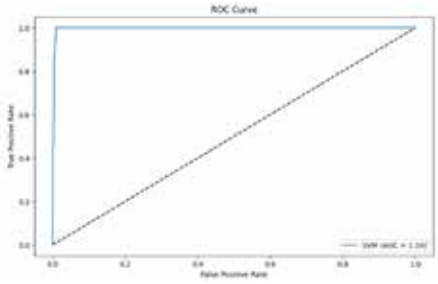

**Figure 21:** ROC curve of SVM. A perfect area under the curve (AUC) of 1.0 indicates that the SVM.

- **(Figure 21) ROC Curve:** The model has an AUC of 1.00, indicating that it is nearly perfectly accurate.
- **(Figure 20) Confusion Matrix:** The model shows excellent predictive performance with negligible errors, making it extremely good for detecting fraudulent transactions.

#### *6.3.2.6 Interquartile Range:*

**Figure 22:** Confusion Matrix of IQR. This figure demonstrates the extremely poor performance of the IQR model, as it has many false negatives. It shows that IQR is performing exceptionally poorly in fraud detection.

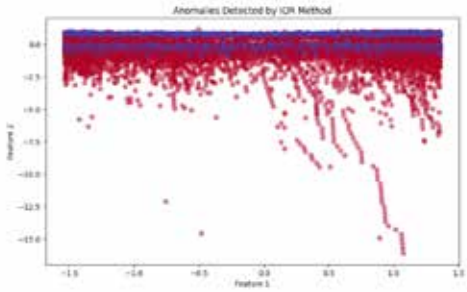

**Figure 23:** Scatter Plot of IQR. By looking at this figure, we can clearly see the poor performance of IQR in anomaly detection as data points are significantly deviating from the main concentration of data (above Feature 1 from -1.0 to 1.5 and Feature 2 around 0) are clearly marked as anomalies that indicate potential issues or exceptional cases within the dataset.

- **(Figure 22) Confusion Matrix:** While the model can detect some fraudulent transactions, i.e., 84, it also has a high false positive rate that shows it incorrectly labels large numbers of legitimate transactions as fraud.
- **(Figure 23) Scatter Plot.:** It visualizes the distribution of the first two features in the datasets. The Blue Dots display legitimate transactions, and the Red Dots show fraudulent transactions.

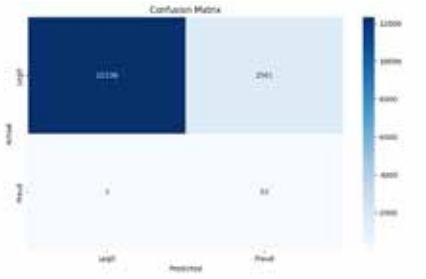

**Figure 24:** Confusion Matrix of Z-Scored. This figure demonstrates the poor performance of the Z-Scored model, as it has a high number of false negatives, 1561. It shows that Z-Scored is performing badly in fraud detection.

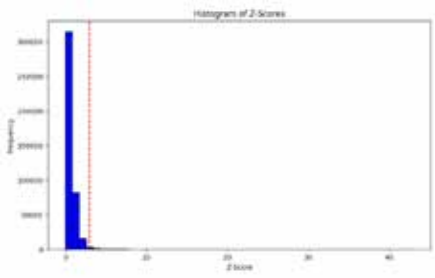

**Figure 25:** Histogram of Z-Scored. The distribution of Z-Scores highlights the concentration of data points within a standard deviation close to zero.

- **(Figure 25) Histogram:** The blue bars indicate the distribution of Z-Scores across the dataset, and the red dashed line represents the Z-Score threshold used to identify anomalies; any Z-Score beyond this threshold is considered an anomaly. Most data points have Z-scores near zero, which indicates they are close to the mean, and very few points have high Z-scores that display anomalies or outliers.
- **(Figure 23) Confusion Matrix:** It correctly identifies legitimate transactions but has a large number of false positives. The number of currently predicted fraud transactions is very low, with few false negatives.

#### *6.3.2.8 Principal Component Analysis:*

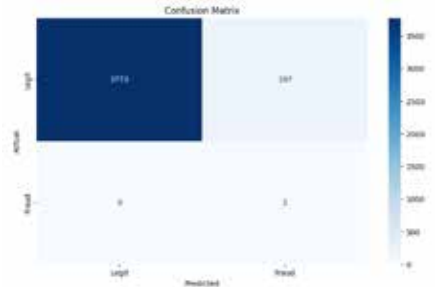

**Figure 26:** Confusion matrix of PCA. This figure demonstrates the excellent performance of PCA as it has a comparatively low number of false negatives, i.e., 197. It shows that PCA is performing relatively well in fraud detection.

**Figure 27:** Cluster Visualization with PCA. The majority of credit card transactions are clustered tightly, with fraudulent transactions (red) sparsely distributed, indicating their rarity and distinct separation.

- **(Figure 27) Scatter Plot:** Most transactions are clustered, and fraud transactions are more scattered.
- **(Figure 23) Confusion Matrix:** The model correctly identifies most legitimate transactions but has false positives, and very few correctly identified fraud transactions.

These results highlight the varying efficiency of different traditional and modern machine learning methods for anomaly detection. In traditional methods like PCA and Z-Score Analysis, errors were more prevalent as they struggled with high-dimensional data and gave high false positives, confirming their limitations' expectations. On the contrary, modern methods like KNN, FCM, and SVM performed well. In contrast, KNN and SVM performed well in supervised learning and FCM in unsupervised learning, which supports the hypothesis of advanced modern machine learning methods.

These results give useful analysis to enhance cybersecurity and fraud detection systems that ensure accurate and reliable anomaly detection. The relationship between variables was evident as modern methods consistently performed better than traditional ones in handling complex datasets. These findings are consistent with existing literature that further validates the superiority of advanced machine learning methods. Some anomalies were noted, such as the high false positive rates in simpler models due to their sensitivity to data distribution and outliers. These experiments can be improved using more extensive and more diverse data sets as they provide a more comprehensive understanding and validate these results across various applications.

## ■ Conclusion

This paper studies traditional and modern machine learning methods for anomaly detection, emphasizing their uses in cybersecurity and fraud detection. Although traditional methods like Principal Component Analysis (PCA) and Z-Score Analysis are simple, they are ineffective in handling high-dimensional and complex data. Nevertheless, modern machine learning methods like Logistic Regression, KNN, K-means, FCM, and SVM demonstrate excellent performance and high accuracy. Our experimental results indicate that KNN and SVM are effective in supervised learning methods, whereas FCM is highly effective and precise in unsupervised learning. Moreover, this research contributes by offering a detailed comparison of various methods, providing guidance for selecting the most appropriate models depending on the application.

Even though Machine learning methods generally outperform traditional methods, they have their specific strengths and limitations. Future research should focus on developing hybrid models that are precise and more advanced, like machine learning methods, and provide the simplicity of traditional methods. Selecting the appropriate method is necessary for enhancing system security and operational integrity, and our study provides a foundation for future exploration and development in the field of Machine Learning and Cybersecurity.

## ■ References


1. Han, Jiawei, Jian Pei, and Hanghang Tong. *Data mining: concepts and techniques*. Morgan kaufmann, (2022).
2. Chouiekh, Alae, and EL Hassane Ibn EL Haj. "Convnets for frau d detection analysis." *Procedia Computer Science* 127 (2018): 133-138.
3. Iakovidis, Dimitris K., Spiros V. Georgakopoulos, Michael Vasilak akis, Anastasios Koulaouzidis, and Vassilis P. Plagianakos. "Detec ting and locating gastrointestinal anomalies using deep learning a nd iterative cluster unification." *IEEE E Transactions on Medical Imaging* 37, no. 10 (2018): 2196-2210.
4. Jagannatha, Abhyuday N., and Hong Yu. "Bidirectional RNN for medical event detection in electronic health records." In *Proceedin gs of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, vol. 2016, p. 473. NIH Public Access, (2016).
5. Ghosh, Sushmito, and Douglas L. Reilly. "Credit card fraud detection with a neural-network." In *System Sciences, 1994. Proceed ings of the Twenty-Seventh Hawaii International Conference on*, vol. 3, pp. 621-630. IEEE, (1994).
6. Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomal y detection: A survey." *ACM computing surveys (CSUR)* 41, no. 3 (2009): 1-58.
7. Krepinevich, Andrew F. *Cyber warfare*. Center for Strategic and B udgetary Assessments, (2012).
8. Scarfone, Karen, and Peter Mell. "Guide to intrusion detection an d prevention systems (idps)." *NIST special publication* 800, no. 200 7 (2007): 94.
9. Patcha, Animesh, and Jung-Min Park. "An overview of anomaly d etection techniques: Existing solutions and latest technological tr ends." *Computer Networks* 51, no. 12 (2007): 3448-3470.
10. Fernandes, Gilberto, Eduardo HM Pena, Luiz F. Carvalho, Joel JPC Rodrigues, and Mario L. Proença Jr. "Statistical, forecasting and metaheuristic techniques for network anomaly detection." In *Proceedings of the 30th Annual ACM Symposium on Applied Comput ing*, pp. 701-707. (2015).



DOI: 10.36838/v6i11.2

11. Jia, Watson, Raj Mani Shukla, and Shamik Sengupta. "Anomaly detection using supervised learning and multiple statistical methods." In 2019, the 18th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1291-1297. IEEE, (2019).
12. Ray, Partha Pratim, and Dinesh Dash. "IoT-Based Small Scale Anomaly Detection Using Dixon's Q Test for e-Health Data." *Applied System Innovation* 4, no. 4 (2021): 100.
13. Zhang, Yi, Peng Peng, Chongdang Liu, and Heming Zhang. " Anomaly detection for industry product quality inspection based on Gaussian restricted Boltzmann machine." In 2019 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1-6. IEEE, (2019).
14. Deepthi, Adathakula Sree. "Anomaly Detection Using Principal Component Analysis." (2014).
15. Issariyapat, Chavee, and Kensuke Fukuda. "Anomaly detection in IP networks with principal component analysis." In *2009 9th International Symposium on Communications and Information Technology*, pp. 1229-1234. IEEE, (2009).
16. Shyu, Mei-Ling, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. "A novel anomaly detection scheme based on principal component classifier." In Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, pp. 172-179. IEEE Press, (2003).
17. Brauckhoff, Daniela, Kave Salamatian, and Martin May. "Applying PCA for traffic anomaly detection: Problems and solutions."In *IEEE INFOCOM* 2009, pp. 2866-2870. IEEE, (2009).
18. Yaro, Abdulmalik Shehu, Filip Maly, and Pavel Prazak. "Outlier Detection in Time-Series Receive Signal Strength Observation Using Z-Score Method with S n Scale Estimator for Indoor Localization." *Applied Sciences* 13, no. 6 (2023): 3900.
19. Iglewicz, Boris, and David Hoaglin. "The ASQC basic references in quality control: statistical techniques." *How to detect and handle outliers* 16 (1993): 1-87.
20. Tukey, John Wilder. *Exploratory data analysis*. Vol. 2. Reading, MA: Addison-Wesley, (1977).
21. Hodge, Victoria, and Jim Austin. "A survey of outlier detection methodologies." *Artificial intelligence review* 22 (2004): 85-126.
22. Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41, no.3 (2009): 1-58.
23. Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41, no.3 (2009): 1-58.
24. Vapnik, Vladimir Naumovich, and Vlamimir Vapnik. "Statistical learning theory." (1998): 1780.
25. George, Annie, and A. Vidyapeetham. "Anomaly detection based on machine learning: dimensionality reduction using PCA and classification using SVM." *International Journal of Computer Applications* 47, no. 21 (2012): 5-8.
26. Hu, Wenjie, Yihua Liao, and V. Rao Vemuri. "Robust anomaly detection using support vector machines." In *Proceedings of the international conference on machine learning*, vol. 6. University Park, PA, USA: Citeseer, (2003).
27. Tian, Jing, Michael H. Azarian, and Michael Pecht. "Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm." In *PHM Society European conference*, vol. 2, no. 1. (2014).
28. Li, Yang, Binxing Fang, Li Guo, and You Chen. "Network anomaly detection based on TCM-KNN algorithm." In *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pp. 13-19. (2007).
29. Heckerman, David. "A tutorial on learning with Bayesian networks." *Innovations in Bayesian networks: Theory and applications (2008)*: 33-82.
30. Junejo, Imran N. "Using dynamic Bayesian network for scene modeling and anomaly detection." *Signal, Image and Video Processing* 4, no. 1 (2010): 1-10.
31. Usama, Muhammad, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha. "Unsupervised machine learning for networking: Techniques, applications and research challenges." *IEEE Access* 7 (2019): 65579-65615.
32. Rui, Chen, Zhang Fengbin, and Xi Liang. "Anomaly detection algorithm based on FCM with improved krill herd." In *Journal of Physics: Conference Series*, vol. 1187, no. 4, p. 042028. IOP Publishing, (2019).
33. Xie, Linquan, Ying Wang, Liping Chen, and Guangxue Yue. "An anomaly detection method based on fuzzy c-means clustering algorithm." In *Second International Symposium on Networking and Network Security (ISNNS, 10) Jinggangshan, PR China*, pp. 89-92. 2010.
34. Münz, Gerhard, Sa Li, and Georg Carle. "Traffic anomaly detection using k-means clustering." In *Gi/itg workshop mmbnet*, vol. 7, no. 9. (2007).
35. Nieves, Jose F., and Yu Cathy Jiao. "Data clustering for anomaly detection in network intrusion detection." *Research Alliance in Math and Science* 1 (2009): 1-2.
36. Kumar, B. Kiran, and A. Bhaskar. "Identifying Network Anomalies Using Clustering Technique in Weblog Data." *International Journal of Computers & Technology* 2, no. 3 (2012).
37. Yang, Kun, Samory Kpotufe, and Nick Feamster. "An efficient one-class SVM for anomaly detection in the internet of things." *arXiv preprint arXiv:2104.11146* (2021).
38. Chen, Yuting, Jing Qian, and Venkatesh Saligrama. "A new one-class SVM for anomaly detection." In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3567-3571. IEEE, (2013).
39. Wang, Zhongfeng, Yatong Fu, Chunhe Song, Peng Zeng, and Lin Qiao. "Power system anomaly detection based on OCSVM optimized by improved particle swarm optimization." *IEEE Access 7 (2019)*: 181580-181588.
40. Elmrabit, Nebrase, Feixiang, Zhou, Fengyin, Li, and Huiyu, Zhou. "Evaluation of Machine Learning Algorithms for Anomaly Detection. "In 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security) (pp. 1-8). (2020).
41. Nassif, Ali Bou, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. "Machine learning for anomaly detection: A systematic review." *Ieee* Access 9 (2021): 78658-78700.


## Authors


Aditya Raj is a rising senior at Aakash International School and wants to pursue higher studies in the USA. Aditya wants to major in CS and AI. Aditya is passionate about technology and has deep knowledge of Machine Learning, Deep Learning, AI, and Software Development.

Sanskar Sharma is a rising senior at Orion CBSE School and wants to pursue Computer Science as a major. Sanskar is passionate about technology and has deep knowledge of Computer Science and Machine learning.


## Appendix

- Dataset Used: Credit Card fraud Detection, https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- Anomaly Detection Prediction Code:

1. KMeans: https://colab.research.google.com/drive/1D-O7SUrNf6mKW2akz9EEs3JE-NQ10anc#scrollTo=5poB2zxuhUId
2. Logistic Regression: https://colab.research.google.com/drive/1C1_c4pUXNGYCj6RqOVPq5ME0U86vLAjr
3. FCM: https://colab.research.google.com/drive/1-YKmrPL3jd

Y_2FSrUSXEaxIOwhQ5SSQZ
4. KNN: https://colab.research.google.com/drive/1XrEFePDvvnWzcTAoBeNLosZpJJEhbHlp
5. Z-Scored: https://colab.research.google.com/drive/1R_89wkWloqncJduXnl6DWyEaqWSMqFSz
6. PCA: https://colab.research.google.com/drive/1nty9eIi2gDwnLn51yeMK_RVwypwDNSUQ
7. IQR: https://colab.research.google.com/drive/1pRlB80hYi8JhAFBCiNnB6xED9SHLF34T
8. SVM: https://colab.research.google.com/drive/1ik5eKSwMbyk0mnm50vkdbbs-3nRowGoV


DOI: 10.36838/v6i11.2