e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

AN IMPROVISED MACHINE LEARNING TECHNIQUES FOR ANOMALY DETECTION

Sushma Kumari*1, Mrs. Anita Ganpati*2

*1,2Department Of Computer Science, Himachal Pradesh University, India.

DOI : https://www.doi.org/10.56726/IRJMETS59828

ABSTRACT

Insurance fraud detection is a challenging problem due to the diversity of fraudulent schemes and the lack of known fraud cases in typical datasets. Developing effective detection models necessitates a balance between minimizing financial losses from fraud and controlling the costs associated with anomalies. This paper addresses the complexities of insurance fraud detection by employing machine learning techniques for anomaly detection namely DBSCAN, Autoencoders, and Isolation Forest. Machine learning, with its ability to learn from data and improve over time, provides significant tools for detecting fraudulent patterns that older methods may not reveal. Effective fraud detection algorithms must find a balance between avoiding financial losses and minimizing the costs associated with anomalies. Anomaly detection, a crucial aspect of machine learning, identifies data points that significantly deviate from the norm within a dataset. Detecting anomalies is crucial for maintaining data integrity and ensuring accurate analysis and decision-making, as anomalies can indicate errors, rare events, or fraudulent activities. This research also explores the enhancement of anomaly detection through the ensemble of various machine-learning techniques for anomaly detection. The ensemble approach combines the strengths of different algorithms to improve overall detection accuracy. In this study, an ensemble-based machine-learning techniques has been proposed for anomaly detection using evaluation parameters such as accuracy, precision, recall, and F1-score. The findings demonstrate that the ensemble approach offers superior anomaly detection capabilities, providing a robust solution for insurance fraud detection.

Keywords: Machine-learning, Anomaly Detection, Isolation Forest, Autoencoder, DBSCAN.

I. INTRODUCTION

The current societal landscape has witnessed a surge in the volume and complexity of information being processed daily. This increased utilization is necessary for effectively managing current industrial processes, relying on data obtained from the processes themselves. This data needs to be cleaned and transformed into usable information for creating meaningful visualizations, inputting into complex control and prediction algorithms, or storing for future reference. Furthermore, the reliability of the data is crucial. Accurate information is essential for obtaining correct responses from managed processes, while incorrect information can lead to inefficiency, loss of precision, and negatively impact the organization's reputation or bottom line.

The data is generally acquired from the process through sensors, manual input, or automated systems. To prevent errors in the data acquisition process, the data is sanitized and, if possible, corrected. This helps to stop errors from spreading throughout the system. Data that cannot be corrected may appear abnormal compared to other values in the dataset or compared to the median of the dataset. In any case, this could indicate either erroneous data or valid data that signals a potential issue with the data acquisition process or the process itself. Therefore, identifying abnormal data is an important indicator of data quality and a valuable aspect of data analysis. [1]. Because there are many different kinds of fraud and few known fraud cases in standard samples, detecting insurance fraud [2] is a challenging problem. Finding a balance between the costs associated with false warnings and the money saved by eliminating losses is critical when developing detection algorithms [3]. Put more simply, there are a variety of ways for people to deceive insurance firms, and there aren't many documented instances of fraud from which to draw lessons. This makes it difficult to detect insurance fraud. Therefore, while developing fraud detection systems, it's critical to minimize costs by preventing fraud and avoiding overspending on false alerts.

The insurance sector in the United States is made up of thousands of businesses that collect trillions of dollars in premiums yearly, and insurance fraud costs more than $40 billion annually. Fraud affects all parties by driving up premium prices, undermining confidence, and impeding creativity and operational efficiency. The insurance sector need cutting-edge solutions to properly identify possible fraud, expedite the processing of valid claims, and thoroughly investigate questionable situations in order to effectively combat fraud [4]. Insurance fraud involves individuals engaging in deceptive activities to gain an advantage from insurance companies. This can include scenarios like hiding incidents not covered by insurance, distorting the facts of an incident, or exaggerating the extent of damage. For instance, someone might lie about a situation not covered by insurance, shift blame to avoid responsibility or inflate the cost of losses [5]. The insurance sector faces challenges dealing with fraud due to its impact on customer satisfaction, delays in payouts, investigation costs, and regulatory pressures. Fraudulent claims not only affect profitability but also set a precedent for dishonest behavior among policyholders.

Machine learning is a transformative field of study that equips computers with the ability to learn and make decisions without being explicitly programmed. It is a branch of artificial intelligence that focuses on the development of algorithms that can adapt and improve from experience. By processing vast amounts of data, machine learning algorithms can identify patterns, make predictions, and uncover insights that would be challenging or impossible for humans to discern manually. Machine learning algorithms have proven especially useful for detecting anomalies. Anomaly detection is the process of identifying data points, events, or observations that differ considerably from the norm. These abnormalities may signal serious problems such as fraud, errors, or other significant events that demand attention. Traditional anomaly detection approaches frequently struggle with the sheer size and complexity of modern datasets. However, machine learning algorithms excel in this area because they use statistical, mathematical, and computational techniques to examine and interpret data. A dataset's usual behavior can be recognized by training machine learning models, such as neural networks, clustering algorithms, and classification algorithms. When these models are trained, they can quickly spot anomalies and distinguish deviations from the norm.

Machine learning-based anomaly detection algorithms are particularly advantageous to the insurance sector. Insurance fraud is a serious problem that costs businesses billions of dollars every year. Conventional rule-based fraud detection systems frequently fall behind the ever-evolving strategies used by con artists. By continuously learning from fresh data and adjusting to new fraud trends, machine learning provides a dynamic solution. Machine learning algorithms are capable of effectively flagging suspicious actions for additional inquiry by sifting through large databases of claims and discovering anomalous patterns or outliers. Additionally, machine learning reduces the need for manual processes by improving anomaly detection efficiency and accuracy. This reduces the frequency of false positives, which may be expensive and time-consuming to examine, and speeds up the process of identifying and addressing abnormalities. Organizations may focus on real problems and reduce risks by automating the detection process and better allocating their resources. Machine learning [6] is a field of study that enables computers to learn without explicit programming. It is an exciting technology that brings computer capabilities closer to human abilities, namely the ability to learn. Machine learning is currently being used in many unexpected places, and its applications continue to grow [7].

Machine learning algorithms can be divided into three categories [8]

Supervised Techniques [9]

Which requires a labeled dataset of "normal" and "abnormal" data and involves training a classifier. However, this approach is not commonly used because of the general unavailability of labeled data and the inherently unbalanced nature of the classes.

Semi-Supervised Techniques [10]

Assume that some of the data is labeled. This could be any combination of the normal or anomalous data. These techniques construct a model that represents normal behavior from a normal training dataset and then test the likelihood of a test instance being generated by the model.

Unsupervised Techniques [11]

Unsupervised approaches are especially useful when the data is unlabeled, which means that there are no predetermined categories or labels assigned to the data points. These techniques are highly appreciated because they can detect hidden patterns, correlations, or structures in data without requiring prior knowledge of the results. Their versatility and robustness make them an essential tool for any data-driven firm looking to get insights and make sound decisions. Unsupervised approaches are indispensable in data science because of their capacity to work with unlabeled data, identify hidden patterns, and adapt to a wide range of applications across domains.

Machine-learning Techniques for Anomaly Detection

Here's a brief overview of the three machine-learning algorithms for anomaly detection that were evaluated:

Isolation Forest

An unsupervised machine learning technique called Isolation Forest [12] is employed to find anomalies. In order to isolate anomalies with fewer partitions, the data is randomly divided into isolation trees. The technique is widely utilized in many different applications, including fraud detection, network intrusion detection, and outlier identification since it is efficient and effective at finding anomalies in huge datasets[13].

Its foundation is the notion that anomalies are simpler to identify than normal points, which was first put forth by Liu et al. in 2008.

The data is divided into isolation trees at random by the algorithm. An isolation tree is a binary tree in which every leaf node denotes an isolated subset of the data, and every internal node denotes a feature and a splitting point on that feature. In order to construct an isolation tree, the algorithm first chooses a random subset of the data, which it then continually divides into subsets by choosing a splitting point and a feature at random until each subset has a single point or the maximum tree depth is reached.

The algorithm calculates the average path length of each point in the isolation trees in order to isolate anomalies. The average number of edges traveled from the root node to the leaf node for a specific place is known as the average path length. Because anomalies require fewer partitions to isolate than normal points, they are distinguished from one another. This is due to the fact that anomalies are distinct from the bulk of the data and are more likely to be identified early on in the partitioning procedure.

Large datasets can be effectively and efficiently analyzed to find abnormalities using the isolation forest algorithm. Its ability to handle high-dimensional data, its capacity to identify anomalies in both sparse and dense areas of the data, and its capacity to identify anomalies that might be clustered or overlapping provide it a number of advantages over other anomaly detection methods.

The average path length of a point in an isolation tree is determined using the isolation forest algorithm using the following formula:

$$h(x) = -E[\log2(w(x))]$$

where w(x) is the path length of the point x in the isolation tree, E[.] is the expected value, and h(x) is the average path length of the point x [14].

The route length of a point in the isolation tree is the number of edges it passes through to go from the root node to the leaf node. The estimated path length of a specific point is calculated by averaging the path lengths of all the isolation trees in the forest. Anomalies differ from regular points in that they have a shorter average path length, indicating that they require fewer partitions to be separated in the isolation trees.

DBSCAN (Density-based Spatial Clustering of Applications with Noise)

A data clustering approach called density-based spatial clustering of applications with noise (DBSCAN) [15] was put forth in 1996 by Xiaowei Xu, Jörg Sander, Martin Ester, and Hans-Peter Kriegel. This non-parametric approach for density-based clustering takes a set of points in space, clusters together the points that are closely packed together (i.e., have many nearby neighbors), and labels the points that are isolated in low-density areas (i.e., whose nearest neighbors are too far away) as outliers. One of the most widely used and frequently quoted clustering methods is DBSCAN.

The algorithm won the 2014 ACM SIGKDD conference's Test of Time Award, which is granted to algorithms that have attracted a lot of interest both theoretically and practically. The follow-up paper "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN" is listed as one of the top eight articles downloaded from the esteemed ACM Transactions on Database Systems (TODS) magazine as of July 2020 [16]. Two parameters are needed for DBSCAN: ε (eps) and minPts, which is the minimal number of points needed to construct a dense region [a]. It begins at a randomly chosen, unexplored starting point. The ε-neighborhood of this point is obtained, and a cluster is initiated if it contains a sufficient number of points. If not, the point is classified as noise. It should be noted that this point could potentially be included in a cluster if it is subsequently discovered in a sufficiently large ε-environment of another point. A point's ε-neighborhood is included in the cluster if it is determined to be a dense component of it. As a result, when points are dense, their own ε-neighborhood is also added, along with all points discovered within the ε-neighborhood. Until the density-connected cluster is fully located, this process is repeated. Next, a fresh, unexplored point is obtained and analyzed, resulting in the identification of an additional cluster or noise [16].

Autoencoder

Kramer initially introduced the autoencoder [17] as a nonlinear generalization of principal components analysis (PCA). The autoencoder is also known as the Diabolo network or autoassociator. It was originally used in the early 1990s. Although dimensionality reduction and feature learning were their most common traditional applications, the idea was eventually extended to the development of generative models for data. The 2010s saw the development of some of the most potent AIs, which used autoencoders layered inside deep neural networks [18]. In order to provide an output that is comparable to the input, AE attempts to learn an approximation of the identity function. The encoder and decoder are its two constituent pieces. The network acquires the skills necessary to effectively compress data (encoder) and reconstruct data into a representation that is similar to the input data (decoder). By figuring out the reconstruction error, AEs are utilized in AE-based anomaly detection to find unusual occurrences. In the field of fraud detection, Schreyer et al. (2017) employed deep autoencoders to find anomalies in large-scale accounting data.

Additionally, a deep autoencoder-based method for novelty identification was presented by Amarbayasgalan et al. (2018). By calculating the error threshold from the deep AE model, their method moves on to a density-based cluster. Next, the compressed data is subjected to density-based clustering in order to obtain low-density novelty groups [19].

II. LITERATURE REVIEW

Abdullah et al. explored the efficiency of various machine learning algorithms in addressing the security challenges posed by NoSQL databases. Their findings indicate that Neural Networks, with a 99.9% accuracy and a 0.2% false positive rate, outperform other methods in detecting anomalies [33].

Hansson et al. modeled insurance claims as sequences of events and applied various deep learning techniques, including Autoencoder (AE), Variational Autoencoder (VAE), COPOD (Copula-based Outlier Detection), and Support Vector Machine (SVM), to learn representations of normal claim sequences. In which AE and Variation AE outperformed with a weighted average F1-Score of 0.93 [2].

Gupta et al. conducted a comprehensive study on the significance of anomaly detection in credit card transactions, especially for fraud detection using Isolation Forest, LOF, and SVM machine learning algorithms for anomaly detection. Their findings concluded that the Isolation Forest algorithm with an accuracy of 99.74% is the most effective for detecting anomalies in credit card transactions [26].

Kersting et al. utilized several algorithms, including Autoencoder, DBSCAN, and Isolation Forest (IF), for anomaly detection in time series data. Their findings indicate that among these, DBSCAN had a lower false alarm rate compared to the other methods, and anomalies detected by DBSCAN were more [45].

Ting et al. proposed the Isolation Forest (iForest) technique for anomaly detection, emphasizing its consistently high AUC scores exceeding 0.9 across datasets, including achieving 0.9999 on the Shuttle and Mulcross datasets. iForest demonstrates superior performance compared to other methods such as one-class SVM and LOF, with AUC values of 0.9999 versus 0.9985 on the Shuttle dataset. It also shows robustness against masking and swamping effects, maintaining AUC values above 0.9 even with up to 50% anomalies [13].

Corizzo et al. explored the effectiveness of various machine learning techniques—Isolation Forest, OCSVM, Autoencoder, and the proposed ensemble model KNN + Autoencoder—in "Spatially-Aware Autoencoders for Detecting Contextual Anomalies in Geo-Distributed Data." Their findings indicate that the KNN + Autoencoder (proposed model) outperformed others with an F1-score of 85% [39].

Bhadri et al. compared the performance of machine learning techniques for anomaly detection including Local Outlier Factor (LOF), Isolation Forest, and Autoencoders. Their findings concluded that Autoencoders outperformed other techniques with a 62% accuracy rate [23].

Maurya et al. evaluated Isolation Forest and Random Forest in "Integrity Shield: Ensuring Real-time Data Integrity in Healthcare IoT with Isolation Forest Anomaly Detection." Their findings concluded that Isolation Forest outperformed, achieving an accuracy of 85% [27].

Bae et al. compared Autoencoder and DBSCAN on the KDD Dataset. Their findings concluded that Autoencoder outperformed, achieving accuracy rates ranging from 84% to 100% [18].

Thimo et al. explored the effectiveness of various machine-learning techniques for anomaly detection. Their findings indicate that OCSVM excels in simplicity and effectiveness, the autoencoder is best for complex scenarios, and DBSCAN is effective in localized anomaly detection. These insights aid in selecting and implementing effective anomaly detection techniques for data-driven process modeling [44].

III. OVERVIEW OF ANOMALY DETECTION

Anomaly detection in data analysis is the identification of rare observations, items, or events that deviate significantly from the majority of data. These anomalies do not conform to normal behavior and may be inconsistent with the rest of the dataset. Anomaly detection has a wide range of applications, including cybersecurity, medicine, machine vision, statistics, neuroscience, law enforcement, and financial fraud detection. Initially, anomalies were identified by their rejection or omission from the data to aid statistical analysis, such as computing the mean or standard deviation. They were also removed to improve the accuracy of models like linear regression. However, anomalies are often of interest and may be the most crucial observations in the entire data set, requiring identification and separation from irrelevant noise or outliers [20].

- An outlier is an observation that deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.
- Anomalies are instances or collections of data that occur very rarely in the data set and whose features differ significantly from most of the data.
- An outlier is an observation (or subset of observations) that appears to be inconsistent with the remainder of that set of data.
- An anomaly is a point or collection of points that is relatively distant from other points in a multi-dimensional space of features.
- Anomalies are patterns in data that do not conform to a well-defined notion of normal behavior [21].

Various definitions have been proposed for an outlier, and there seems to be no universally accepted definition. However, we will use the definition given by Grubbs (Grubbs, 1969), as quoted in Barnett & Lewis (Barnett and Lewis, 1994):

An outlier is an observation that appears to deviate significantly from other members of the sample in which it occurs.

In statistics, outliers refer to observations in a dataset that seem inconsistent with the rest of the data. Barnett and Lewis (1994) describe outliers as observations or subsets of observations that are clearly isolated and different from the main cluster of points.

According to John (1995), outliers can be surprising veridical data, meaning a point that belongs to class A but is actually situated inside class B, making its true classification surprising to the observer.

Aggarwal and Yu (2001) note that outliers can be considered as noise points lying outside a set of defined clusters. Alternatively, outliers can be defined as points that lie outside the set of clusters but are also separated from the noise. These outliers behave differently from the norm [10].

Types of Anomalies

Anomalies in data can be categorized into three types: point, contextual, and collective anomalies, each having unique characteristics and requiring different approaches for detection and management:

Point Anomalies

A single data point that is significantly different from the rest of the dataset. Point anomalies refer to individual data points that significantly deviate from the remaining data. These outliers can be detected by analyzing each data point individually and determining if it lies outside the normal range or data distribution. Point anomalies are the simplest to identify and can be detected using statistical methods.

Contextual Anomalies

Data points that are unusual in a specific context or subset of the dataset. Contextual anomalies refer to data points that appear anomalous only in certain contexts, and not necessarily when examined individually. Detecting these anomalies can be difficult as it requires an understanding of the underlying contextual information. Time series data is particularly susceptible to contextual anomalies as the context is provided by the relationships between neighboring data points.

Collective Anomalies

Patterns in the dataset that deviate significantly from what is expected. Collective anomalies refer to groups of data points which are considered anomalous when observed together, but may not be detected when examining individual data points. These patterns require analysis of the data as a whole group in order to identify the anomalous behaviour. Time series data is particularly susceptible to collective anomalies, where groups of data points may deviate from their regular temporal behaviour.

Multivariate anomalies

Multivariate anomalies refer to data points that appear anomalous when considering relationships between at least two features. These anomalies may not be visible when analyzing each feature independently. Detecting them can be challenging in real-world data as they may be hidden among relationships between multiple variables. However, it is especially important to identify them in industrial settings where interactions between process variables can have a significant impact on the overall performance of the industrial process.

IV. COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR ANOMALY DETECTION

A comprehensive comparative study of machine-learning techniques for anomaly detection is undertaken by thoroughly investigating a wide array of sources, including thorough reviews of research papers, articles, books, and online resources. This meticulous examination of existing literature and implementation of various anomaly detection algorithms led to significant insights. Table 1, below presents a detailed comparative analysis of the machine-learning techniques for anomaly detection:

Table I

Table 1: Comparative analysis of machine-learning techniques for Anomaly Detection: [22] [23] [24] [5] [25] [26] [27] [28] [8] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [17] [39] [40] [41] [7] [42] [2] [43] [44] [45] [46] [47]

Authors Name	Techniques Name	Dataset Used	Best performed Technique	Key Findings
Xin et al.	- (OCSVM) - Isolation Forest - KNN - LOF - Deep ensemble method	-. Decentralized Application (DApp) monitoring data - SMD (Server Machine Dataset) - Vichalana	Deep ensemble Method	ARP(Average Recall and Precision)_Score of 5.1821
Bhadri et al.	- Local Outlier Factor (LOF) - Isolation Forest - Autoencoders	- breast cancer dataset -coronavirus dataset -heart disease	Autoencoder	Accuracy rate of 62%

		dataset		
Diro et al.	-KNN -Random Forest -Decision Tree -ANN -Logistic regression method -SVM and others	- N-BaIoT -CICIDS 2017 -AWID -UNSW-NB15 -NLS-KDD -Kyoto -KDD CUP 1999	ANN	Accuracy rate of 99.4%
Liu et al.	-LOF TDNNR (Time delay neural network regression) -Structured Autoencoder - Autoencoders	Data was collected using 151 sensors.	Structured Autoencoder	Reduce anomaly detection misclassification error by up to 64%.
Ozkum et al.	-Isolation forest -OCSVM -Autoencoders	Credit Card Transaction	OCSVM	F1-Score of 81%
Gupta et al.	-Isolation Forest	Credit Card Transaction	Isolation Forest	Accuracy rate of 98.72%
Maurya et al.	- Isolation Forest -Random Forest	Healthcare Data	Isolation Forest	Accuracy rate of 85%
Haji et al.	- SVM -Random Forest -KNN -Decision Trees Naive Bayes -Neural Networks (including deep learning approaches)	IOT network traffic data, sensor data, device behavior logs, and Synthetic or Simulated Datasets	Random Forest and KNN	Accuracy rate of 99%
Bouman et al.	-KNN -Extended Isolation Forest - k-th Nearest Neighbor (k-thNN)	Real-valued, multivariate, tabular data	Extended Isolation Forest and KNN	With mean AUC-Score 0.770 and 0.737
Petrariu et al	-kNN -LOF -CBLOF -HBOS -Local correlation Integral (LOCI)	small and medium-sized software enterprise	HBOS	Accuracy rate of 98.64%

Das et al.	-OCSVM -IF -LOF -MCD (Minimum Covariance Determinant)	physiological datasets	MCD	Accuracy rate of 84%
WyWiol et al.	-OCSVM -HBOS -ARIMA (Auto Regressive Integrated Moving Average) -Autoencoder	Servo Motor Time-Series Dataset from PROTOS M5e	ARIMA	Detects fault in 0.50 seconds and F1-Score of 100%
Demestichas et al.	- LOF -CBLOF -HBOS -KNN -MCD -PCA -ABOD -Isolation Forest -Auto-encoder	-arrhythmia -letter -mnist -pendigits -satellite	Autoencoder	AUC 99% and Precision 94%
Abdullah et al.	-Logistic Regression -Isolation Forest -Neural Network -Streaming Clustering - Adversarial Drift Detection	MongoDB	Neural Network	Accuracy rate of 99.9%
Vismari et al.	-k-means -SOM (Self-Organizing Maps) -Auto-encoder	Railway System Operational Data	Autoencoder	Accuracy rate of 99.28%
Wilmet et al.	- CNN -Auto-encoder, and - Generative Adversarial Networks (GANs)	-Toothbrush -Bottle -Screw - Leather, and -Transister	Autoencoder	F1-score 87%
Xin et al.	- OCSVM - Isolation Forest - KNN - LOF - Deep Ensemble	decentralized application (DApp) monitoring data	Deep ensemble	Accuracy rate of 87%

Chahla et al.	k-means -ARIMA -Auto-encoders -k-means+LSTM	Dataport	k-Means + LSTM	Accuracy rate of 89%
Zangrando et al.	Isolation Forest -OCSVM -LOF -Neural Network Models	quasi-periodic energy consumption data	Isolation Forest	F1-Score higher than 0.9
Tien et al.	Isolation Forest -OCSVM -Auto-encoder	IOT sensor Data	Autoencoder	Accuracy rate of 97.6%
Corizzo et al.	Isolation Forest -OCSVM -Auto-encoder -Proposed model, k-NN + autoencoder	Geo- Distributed Data	KNN + Autoencoder	F1-score of 85%
Fernandes et al.	-OCSVM -LOF -Elliptical Envelope -Autoencoder with feedforward and LSTM architectures	Multivariate time series data 3W	LOF	F1-Score 91.5%
Rezapour et al.	-OCSVM -Autoencoder -Robust Mahanabolis	Credit Card	Robust Mahanabolis	Robust Mahanabolis performed well
Sharmila et al.	-LOF -IF(Isolation Forest)	Credit Card Transactions	IF	IF demonstrated effective performance
Falcão et al.	- Clustering based - Neighbor-based - Density-based - Statistical- based -Angle-based -Classification-based	KDD Cup 1999	Classification-based	Accuracy rate of 99.7%
Hansson et al.	-Autoencoder (AE) Variation based on LSTM -OCSVM -COPOD (Copula-Based Outlier Detection)	Hedvig Insurance Company dataset	Autoencoder	F1-Score of 93%

Princz et al.	-DBSCAN -OCSVM -IF -LOF -AE -k-Means	FESTO FMS 50 didactics system data	OCSVM	F1-Score of 87%
Schindler et al.	-OCSVM -DBSCAN -Autoencoder	Process Dataset	Autoencoder	AUC_ Mean 0.681
Kerstinga et al.	-Autoencoder -DBSCAN -IF	Time series data	DBSCAN	Anomalies detected by DBSCAN is more
Filiz et al.	DBSCAN	Time series data	DBSCAN	DBSCAN finds anomalies in both moderate and extreme value ranges
Kevin et al.	DBSCAN	Flight Data	DBSCAN	Anomalies detected within 100-200 seconds

Several machine learning techniques were evaluated and a thorough comparative analysis of different anomaly detection algorithms was carried out by looking through a large number of research articles. The objective was to find the top-performing algorithms for several elements of anomaly detection, including efficacy and accuracy in detecting anomalies.

As shown in above Table I, in most cases, the Autoencoder and Isolation Forest machine learning techniques are widely used. These machine learning techniques for anomaly detection performed better at identifying anomalies. This is mostly because of its capacity to pick up intricate data representations and patterns. The autoencoder performed better than the other models in terms of accuracy. This indicates that it minimized false positives, or normal data that was mistakenly identified as anomalies, and was more reliable in correctly identifying abnormal data points. In order for autoencoders to function, the data must first be compressed into a lower-dimensional representation and then rebuilt. High reconstruction errors are indicative of anomalies since these data points do not fit well into the taught usual patterns. Isolation Forest excels in detecting anomalies by taking advantage of anomalies' tendency to be isolated in the feature space, making it a reliable alternative for anomaly identification across multiple domains and data types. When it comes to identifying anomalies in datasets where the anomalies are not easily distinguished from regular data points, DBSCAN proved to be highly useful. In order to detect anomalies—points that do not belong to any cluster—DBSCAN clusters data points according to density. For datasets with variable densities and uneven distributions, this approach is quite helpful. DBSCAN is a flexible tool for a range of applications since it performs exceptionally well in spatial and density-based anomaly detection settings.

Autoencoder and Isolation Forest machine learning techniques for anomaly detection are widely used and have outperformed other techniques in most cases. Additionally, DBSCAN is useful for detecting most anomalies, as shown in Table 1. In this study, these machine learning techniques are employed for improved anomaly detection.

V. PROBLEM STATEMENT

Detecting fraudulent vehicle insurance claims is critical for insurance firms to avoid financial losses and maintain trust with policyholders. Conventional rule-based systems frequently fail to adapt to developing fraud tactics, resulting in inefficiencies and an increase in false positive anomaly detection. This study tries to overcome these issues by focusing on advanced machine-learning techniques- Autoencoder, Isolation Forest, and DBSCAN, designed specifically for anomaly identification in vehicle insurance data. The study aims to determine the most successful ways for detecting abnormalities in insurance claims by thoroughly studying and comparing several machine learning techniques, which range from traditional statistical methods to recent ensemble techniques and deep learning models. Furthermore, the study proposes improvements to existing techniques to improve their accuracy, precision, and recall in detecting fraudulent claims. This work intends to give insurance businesses strong tools to improve fraud detection capabilities, protect financial assets, and maximize claim processing efficiency through theoretical analysis and empirical evaluation using real-world datasets.

The specific objectives are:

1. To study and analyze various machine learning techniques for anomaly detection.
2. To evaluate the effectiveness of machine learning techniques in detecting anomalies, and evaluate the best-performing one using empirical methods.
3. To propose an enhanced anomaly detection technique and assess its performance against existing techniques.

VI. RESEARCH METHODOLOGY

This study utilized machine learning algorithms for anomaly identification, employing Isolation Forest for unsupervised isolation, Autoencoder for neural network-based encoding efficiency, and DBSCAN for clustering and outlier detection. The Vehicle Insurance Fraud Detection dataset, available on Kaggle, underwent preprocessing including cleaning, normalization, and partitioning into training, validation, and testing sets. Dataset management and integration with Python-based machine learning algorithms were handled using CassandraDB version 3.11.4 and scikit-learn within Python version 3.12.2. Anaconda version 2.5.2 served as the integrated development environment for implementing algorithms. Model robustness was ensured through cross-validation techniques implemented with TensorFlow and PyTorch. Anomaly detection tools included Python for programming, scikit-learn for Isolation Forest and DBSCAN implementations, and TensorFlow and PyTorch for Autoencoder models. The Cassandra-driver library facilitated connectivity between CassandraDB and Python for seamless integration with machine-learning workflows.

Proposed Technique for Machine Learning Techniques for Anomaly Detection

The proposed architecture employs several advanced machine-learning techniques to detect anomalies in an insurance database. The system uses an Autoencoder neural network, Isolation Forest, and DBSCAN clustering to identify and analyze fraudulent claims and unusual patterns in vehicle insurance records.

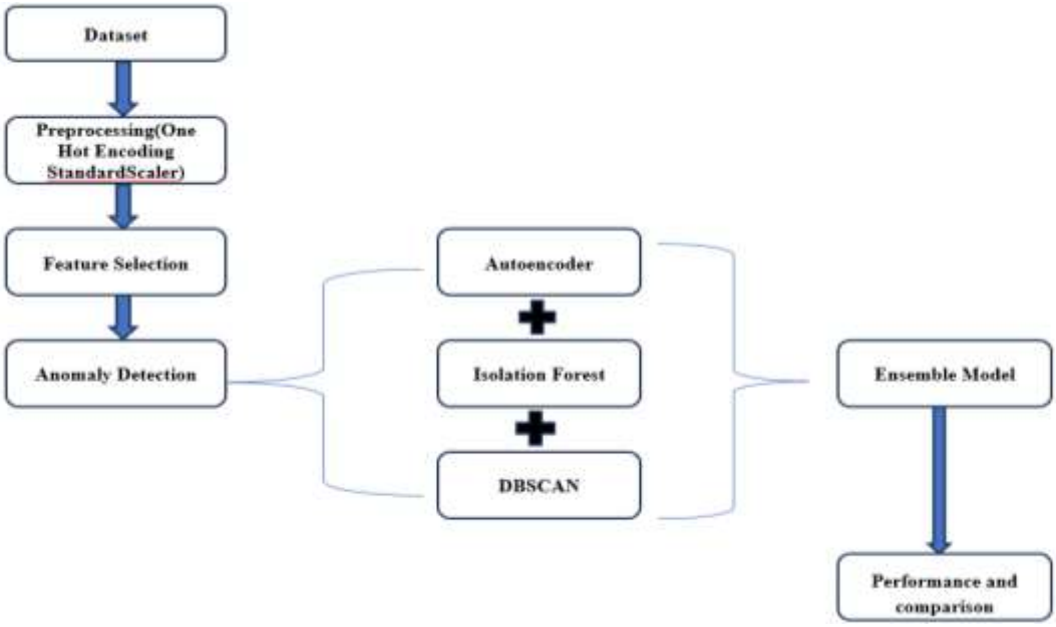

Figure 1: Proposed System Architecture for Anomaly Detection

The datasets are identified in the proposed system architecture shown in Figure 1. After the dataset has been discovered, the data is preprocessed and used to determine the dataset's correlation. The DBSCAN, Autoencoder, and isolation forest algorithms are applied to this dataset in order to identify any anomalies. An 8:2 ratio is employed for training and testing the dataset, meaning that 80% of the data is used for technique training and 20% is used for testing. Accuracy is produced by applying the machine-learning techniques for anomaly detection to random data samples. These methods are used on both preprocessed and raw data. Finally, the accuracy, recall, precision, and F1 scores of the three outcomes will be compared together.

1) Dataset Description

Vehicle insurance fraud is a system in which individuals conspire to make false or inflated claims for property damage or personal injuries following an accident. Common strategies include staging accidents, in which fraudsters intentionally cause crashes; phantom passengers, in which people who were not there at the accident claim injuries; and inflating personal injury claims far beyond actual injuries.

In order to detect fraudulent claims, comprehensive data regarding vehicle insurance claims is included in the "Vehicle Insurance Claim Fraud Detection" dataset on Kaggle [48]. The dataset has 33 features, 15,421 entries, and information about vehicles, accidents, policies, and a fraud indicator (the target variable "FraudFound_P"). This dataset is commonly used as a baseline for creating and testing machine learning models designed to detect vehicle insurance fraud. Its broad use is evidenced by several research projects and studies, as well as a large number of views (88.2K) and downloads (10.9K) on Kaggle.

2) Data Preprocessing

One data mining technique is data preparation, which is converting unprocessed data into a comprehensible format. Real-world data is likely to contain a high number of inaccuracies and is frequently inconsistent, deficient in specific behaviors or trends, and/or incomplete. Preprocessing data is a tried-and-true way to address these problems. Preprocessing data gets unprocessed data ready for additional processing. Applications that rely on databases, like customer relationship management, and rule-based systems, like neural networks, employ data preprocessing. Due to the unevenness of the data, the values could be higher or lower [23].

3) One Hot Encoding Standard Scaler

The process of converting categorical information into numerical features that may be fed into algorithms for machine learning and deep learning is known as one hot encoding. A binary vector represents a categorical variable in this way: all values in the vector would be 0, with the exception of the i^{th} value, which would reflect the variable's i^{th} category and be represented by 1. The length of the vector is equal to the number of unique categories in the variable [49].

4) Feature Selection

Only a small number of the dataset's variables are needed to build the machine learning model; the remaining features are either redundant or unimportant. The accuracy and general performance of the model may be adversely affected if we fill the dataset with too many unnecessary and redundant characteristics. Therefore, it is crucial to find and pick the best features from the data and eliminate any unnecessary or unimportant information. Feature selection in machine learning helps with this process. One of the key ideas in machine learning, feature selection has a significant effect on the model's performance. Because machine learning is based on the "Garbage In, Garbage Out" theory, we must always feed the model with the most relevant and appropriate dataset in order to improve the outcome .

5) Ensemble Anomaly Detection Model

Ensemble learning [50] is a machine learning technique that solves a problem by training several learners. Unlike traditional machine learning methods, which learn a single hypothesis from training data, ensemble approaches aim to construct a group of hypotheses and combine them to form a new hypothesis. Most ensemble techniques use a single base learning algorithm to generate what are known as homogeneous base learners. However, other approaches employ numerous learning algorithms and are known as heterogeneous learners. The basic goal of ensemble learning is to increase the performance of a model by merging many learners [51]. Normally, ensembles are created in two steps. Initially, many base learners are created, which are then integrated. Several combination strategies are utilized. For anomaly classification, majority voting is a popular combination strategy. The ultimate choice in this study is determined by majority voting, which requires the agreement of more than half of the base learners. To analyze the performance of an ensemble model that combines Autoencoder, DBSCAN, and Isolation Forest approaches to detect abnormalities in NoSQL database systems. The goal is to combine the strengths of each technique to obtain higher detection accuracy and dependability.

Evaluation Parameters for Performance [52]

Precision

The percentage of cases that the model flags as anomalies that are, in fact, genuine anomalies out of all instances projected to be anomalies is known as precision. The quantity of anomalies that are accurately classified as anomalies is known as True Positives (TP). The quantity of non-anomalies that are mistakenly categorized as anomalies is known as False Positives (FP). High accuracy reduces false alarms by indicating that the model is very likely to be true when it predicts an abnormality. Ensuring that detected anomalies are, in fact, exceptional occurrences that call for additional research or action is a crucial parameter in anomaly detection. Precision refers to the fraction of predicted positives correctly categorized as positive, defined as

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

Recall

Recall is important in anomaly identification since it measures how well the model captures all actual anomalies in the dataset. A high recall indicates that the model is effective at identifying the majority of the anomalies present, reducing the likelihood of missing potentially crucial occurrences. Recall commonly known as true positive rate (TPR), is defined as the fraction of all positive samples that were correctly categorized as positive, which,

$$\textbf{Recall} = \textbf{TPR} = \frac{TP}{TP + FN}$$

F1-Score

The F1-score is a single metric that balances precision and recall, offering a comprehensive assessment of a model's ability to detect anomalies. It is computed as the harmonic mean of precision and recall:

$$\textbf{F1} = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

The F1-score varies between 0 and 1, where:

A score closer to 1 implies great precision and recall, implying that the model is able to find abnormalities with high accuracy and thoroughness. A score close to zero suggests low precision, recall, or both. The F1-score is especially valuable in anomaly detection because it provides a fair evaluation of the model's ability to correctly categorize anomalies while avoiding false positives and false negatives. It is frequently employed as the major evaluation criterion, alongside precision and recall, to gauge overall model effectiveness.

Accuracy

The most common of which is accuracy, which is defined as the fraction of correct predictions made by the model. Formally defined as

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy assesses the overall correctness of the model's predictions in both normal and abnormal situations.

Where, True Positives (TP) are anomalies that are accurately identified as such. True Negatives (TN) are normal situations that have been appropriately identified as normal. Total Predictions equals the sum of True Positives, True Negatives, False Positives, and False Negatives. Accuracy is a generic indicator of how well the model distinguishes between typical and unusual instances. However, in highly imbalanced datasets with few anomalies (e.g., fraud detection), accuracy may not provide a clear view of the model's performance. Other metrics such as precision, recall, and F1-score should be considered, with a particular emphasis on anomaly detection performance in terms of successfully recognizing and minimizing anomalies.

VII. RESULTS

Anomaly Detected by Machine Learning Algorithms: Autoencoder, Isolation Forest, and DBSCAN

In many different fields, anomaly detection is essential for spotting odd trends or anomalies that are critical to preserving system security and integrity. Machine learning techniques have shown to be highly effective, providing sophisticated methods for identifying anomalies within large and intricate datasets. The implementation of three well-known machine-learning techniques for anomaly detection—Autoencoder, Isolation Forest, and DBSCAN—is the main emphasis of this work. Every technique has its own advantages. Complex data patterns are best captured by Autoencoder, anomalies are efficiently isolated by Isolation Forest, and anomalies are identified by DBSCAN using density clustering.

a) Autoencoder

The Autoencoder is a neural network that uses input data to recreate itself. It learns to compress and decompress information effectively through training on typical data. Anomalies are identified by their significant reconstruction error, which indicates deviations from regular patterns in the dataset.

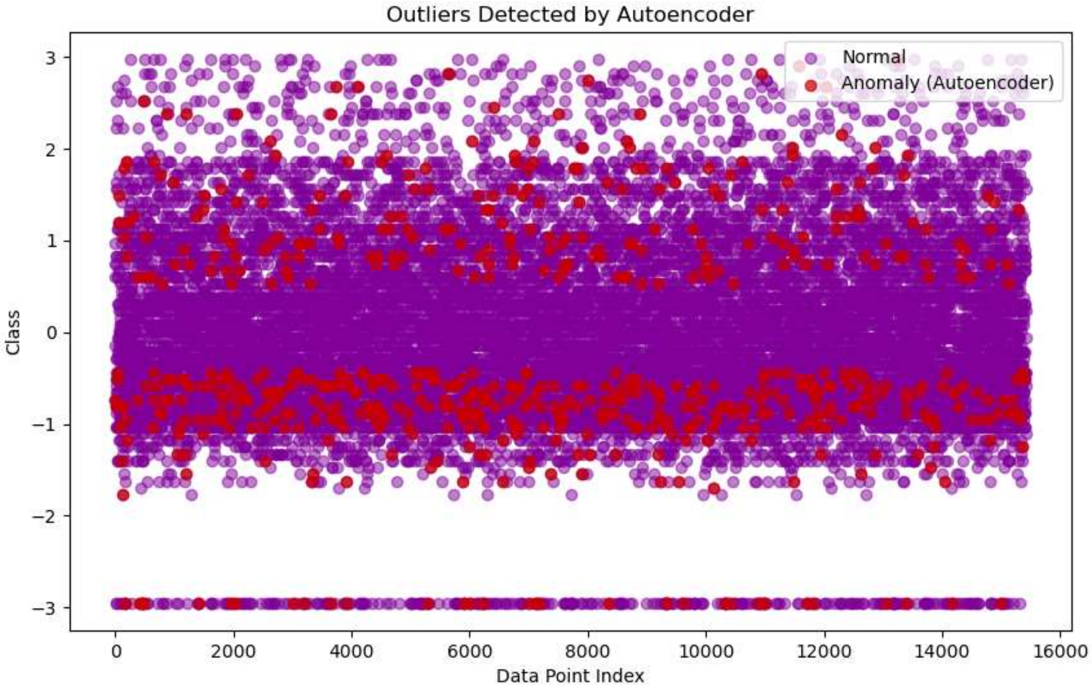

Figure 2: Anomalies detected by Autoencoder

Figure 2 shows anomalies detected by the Autoencoder, where red dots represent anomalies and purple dots represent normal values in the dataset. In our dataset, anomalies were detected in 3.93% of cases by the Autoencoder.

b) Isolation Forest

By building trees, Isolation Forest isolates observations within the data; the path length needed to reach each point determines how isolated it is. Anomalous points are those having shorter routes. Because it is easy to isolate anomalies according to their path lengths inside tree topologies, this method works especially well for finding anomalies in high-dimensional datasets. Isolation Forest proved useful in identifying anomalies in our investigation, demonstrating how it may be used to identify unique data points that differ significantly from the majority.

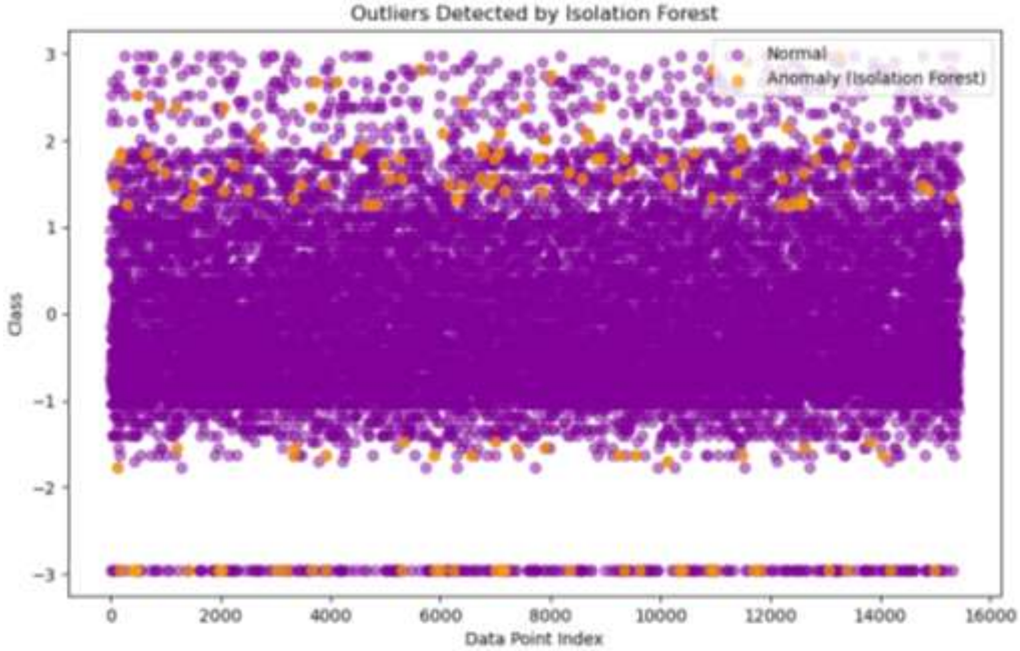

Figure 3: Anomalies Detected by Isolation Forest

Figure 3 shows anomalies detected by the Isolation Forest, where yellow dots represent anomalies and purple dots represent normal values in the dataset. In our dataset, anomalies were detected in 0.95% of cases by the Isolation Forest.

c) DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN classifies points that do not belong to any cluster as anomalies and uses density to discover clusters within the data. This technique works very well with datasets that have a wide range of densities and asymmetric forms. DBSCAN demonstrated its capacity to detect anomalies and unexpected data points that do not fit into the predefined clusters by successfully detecting abnormalities in our analysis.

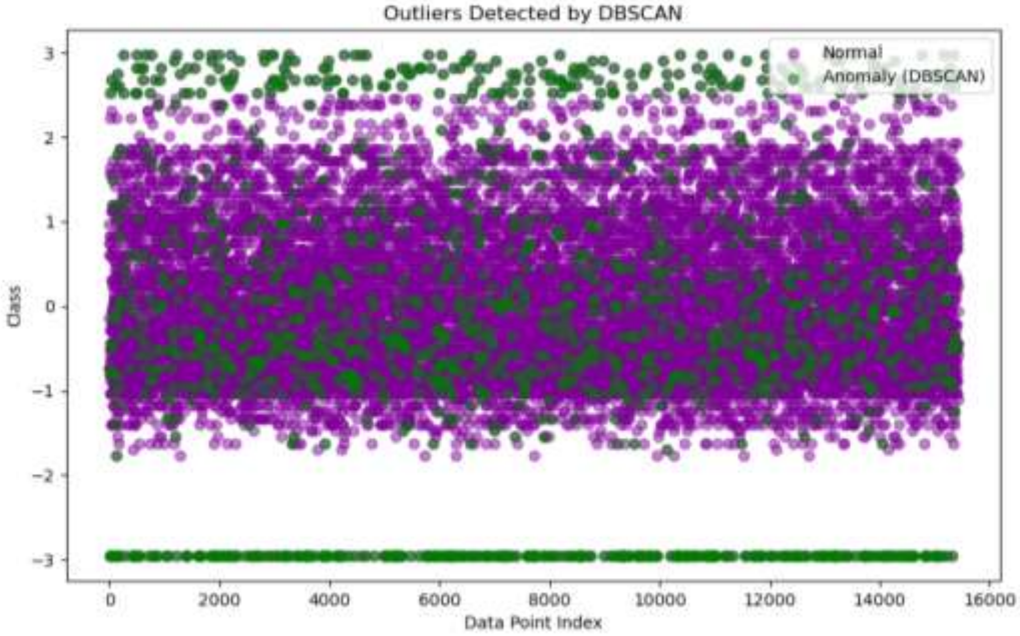

Figure 4: Anomalies detected by DBSCAN

Figure 4 shows anomalies detected by the Autoencoder, where green dots represent anomalies and purple dots represent normal values in the dataset. In our dataset, anomalies were detected in 9.22% of cases by the DBSCAN.

Anomaly Detected by Improvised Machine Learning Techniques

To leverage the strengths of Autoencoder, Isolation Forest, and DBSCAN, an improvised machine learning technique incorporating majority voting is implemented. In this approach, a data point is classified as an anomaly if at least two out of the three methods identify it as such. This voting mechanism effectively mitigates biases and weaknesses inherent in individual techniques, thereby enhancing the accuracy of anomaly detection.

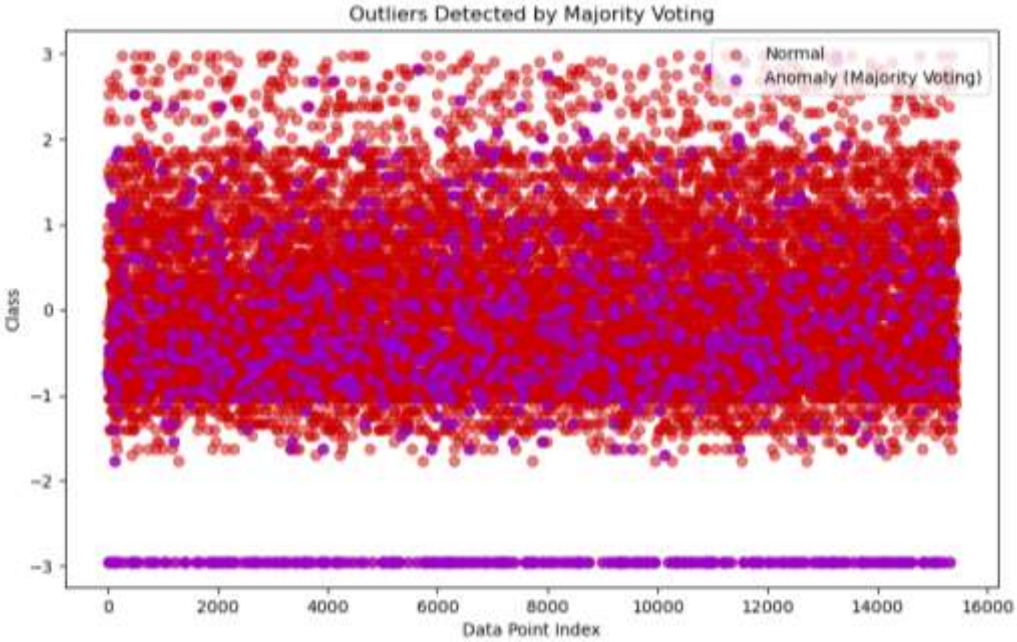

Figure 5: Anomalies Detected by Ensemble Method

An improvised machine learning techniques enhances anomaly detection by integrating the results from Isolation Forest, DBSCAN, and Autoencoder, resulting in a more comprehensive detection set compared to individual methods. This approach improves overall performance by reducing the likelihood of false positives and false negatives, thereby establishing a more robust and reliable detection system. It proves particularly effective for complex datasets with diverse patterns and densities, ensuring that significant anomalies receive appropriate attention without being overlooked.

Comparative results of Machine Learning Techniques for Anomaly Detection: Autoencoder, Isolation Forest, and DBSCAN

The evaluation parameters for three different machine learning techniques for anomaly detection—Autoencoder, Isolation Forest, and DBSCAN—are presented. These techniques were assessed based on their ability to correctly identify normal (0) and anomalous (1) instances. The evaluation parameters include Precision, Recall, F1-Score, and Accuracy. These metrics provide a comprehensive view of the performance of the machine-learning technique, highlighting their strengths and weaknesses in detecting anomalies. To assess the efficiency of anomaly detection algorithms, we used a variety of parameters, including precision, recall, F1-score, and accuracy. Table II shows a full breakdown of the results.

Table 2

Table 2: Comparative results of Machine Learning Techniques for anomaly detection

Evaluation Parameters	Autoencoder (0)	Autoencoder (1)	Isolation Forest (0)	Isolation Forest (1)	DBSCAN (0)	DBSCAN (1)
Precision	0.98	1.00	0.95	1.00	1.00	0.65
Recall	1.00	0.66	1.00	0.16	0.97	1.00
F1-Score	0.99	0.79	0.97	0.27	0.98	0.78
Accuracy	0.98		0.95		0.97	

Table 2 shows an evaluation parameter by comparing different techniques for spotting unusual events. Autoencoder had very accurate results with both high Precision (0.98 for one class and 1.00 for another) and Recall (1.00 and 0.66), which measures how well it finds all relevant instances. Isolation Forest showed good Precision (0.95 and 1.00) but only moderate Recall (1.00 and 0.16), so it was better at being precise with its findings but missed some anomalies. DBSCAN had perfect Precision (1.00 and 0.65) and Recall (0.97 and 1.00), making it consistent in both finding anomalies and being precise about them. The performance parameters indicate that while Autoencoder and DBSCAN provide balanced performance across both classes, the Isolation Forest struggles significantly with detecting anomalies (class 1) despite its high performance for normal instances (class 0). The Autoencoder's high precision and F1-Score for the normal class make it a reliable choice for scenarios with a higher proportion of normal instances, whereas DBSCAN's balanced performance across both classes makes it a versatile choice for varied anomaly detection tasks.

Figure 6 illustrates evaluation parameters precision, recall, and f1-score for each machine-learning technique for anomaly detection, with separate evaluations for both normal (0) and anomalous (1) classes. As shown in Figure 6, the evaluation parameters for each machine-learning technique for anomaly detection are plotted on the y-axis, with the model and class type combinations on the x-axis. The F1-Score is represented by the gray line, Recall by the orange line, and Precision by the blue line.

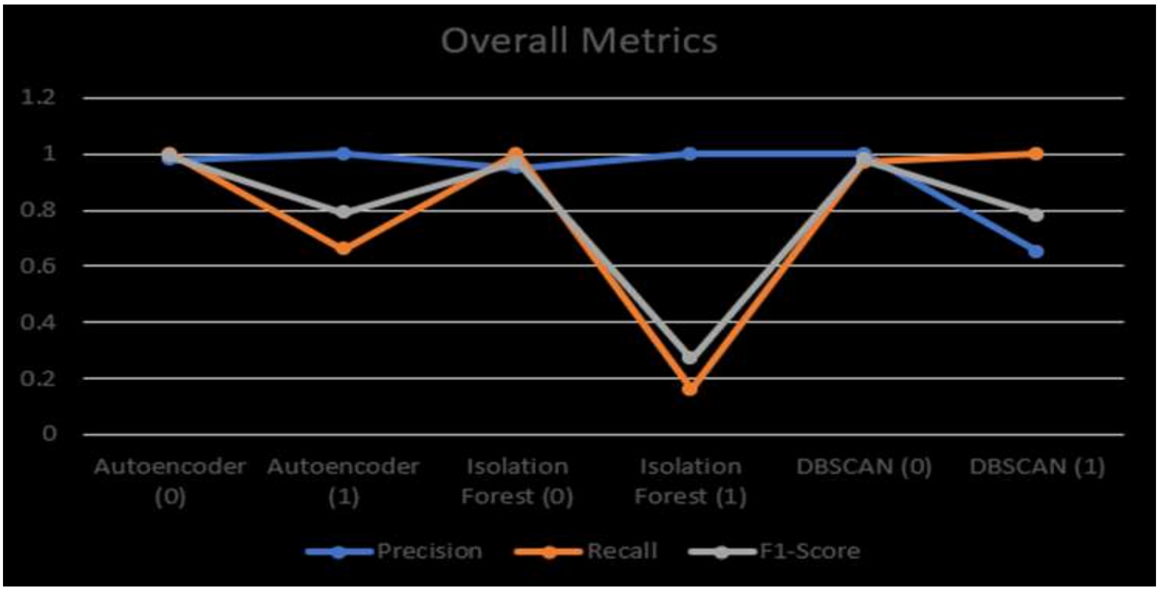

Figure 6: Autoencoder, Isolation Forest, and DBSCAN evaluation parameters line graph

The graph Figure 6, shows that Autoencoder (1) and DBSCAN (0) had the best overall performance, with nearly flawless precision and recall. In contrast, Isolation Forest (1) had a large loss in recall, impairing its total performance despite its excellent precision. This comparison focuses on each algorithm's strengths and limitations in terms of precision, recall, and F1 score.

Analysis of the graphs, figure 6 reveals clear strengths and weaknesses for each machine-learning technique for anomaly detection in terms of precision, recall, and F1-score, providing valuable insights into their suitability for different anomaly detection scenarios. The Autoencoder shows strong precision for both normal and anomalous instances, indicating its reliability in identifying anomalies accurately. However, it has a lower recall for anomalies, suggesting it might miss some anomalous cases. Isolation Forest excels at identifying normal instances but may struggle with anomalies. DBSCAN demonstrates balanced performance across all parameters, making it a versatile choice for anomaly detection tasks.

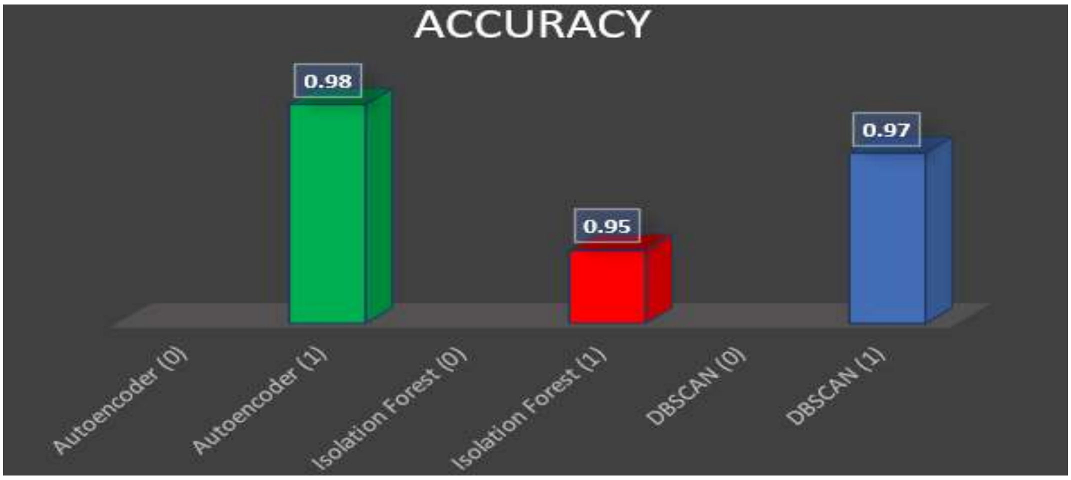

Figure 7: Autoencoder, Isolation Forest, and DBSCAN accuracy bar graph

Figure 7, shows the overall accuracy of the following anomaly detection algorithms: Autoencoder (0), Autoencoder (1), Isolation Forest (0), Isolation Forest (1), DBSCAN (0), and DBSCAN (1). It demonstrates that Autoencoder (0) has the highest accuracy, whereas Isolation Forest (1) has the lowest. The presented parameters show that the Autoencoder outperforms the other methods, followed by DBSCAN. The Isolation Forest, while beneficial, has significant weaknesses in detecting the minority class, as evidenced by its lower recall and F1-score for class 1.

Results of the Improvised Machine Learning Techniques for Anomaly Detection

Autoencoder, Isolation Forest, and DBSCAN provide different advantages that are driving the change from traditional to improvised machine learning techniques in anomaly detection. As seen in Table II, Autoencoder excels in identifying complicated data patterns, allowing it to detect irregularities with high accuracy and few false positives. However, it may suffer recall challenges and miss anomalies in large datasets. Isolation Forest effectively separates abnormalities inside data partitions, resulting in precise anomaly identification. In contrast, DBSCAN employs density clustering to completely identify anomalies, resulting in high recall but occasionally lower precision, resulting in more false positives in specific cases. Together, these strategies address many aspects of data structure and anomaly characterization, hence improving anomaly detection capabilities and accuracy in real-world applications.

The following table III shows the performance parameters for improvised machine-learning techniques for anomaly detection. Two ensemble configurations, Ensemble (0) and Ensemble (1) are evaluated in terms of Precision, Recall, F1-Score, and Accuracy.

Table 3

Table 3: Evaluation parameters of improvised machine learning technique for Anomaly Detection

Evaluation Parameters	Ensemble (0)	Ensemble (1)
Precision	1.00	0.76
Recall	0.98	1.00
F1-Score	0.99	0.86
Accuracy	98.13	

Table 3 shows the evaluation of improvised machine-learning techniques for anomaly detection across two categories (0 and 1). It achieved a perfect Precision of 1.00 for class 0 and a solid Precision of 0.76 for class 1, with Recall rates of 0.98 for class 0 and 1.00 for class 1. The F1-Scores were 0.99 for class 0 and 0.86 for class 1, demonstrating strong overall performance. Table 2 shows how the evaluation parameters highlight the ensembles' strong performance in accurately finding abnormalities, with different precision and recall trade-offs. Ensemble (0) delivers near-perfect precision and strong recall, making it ideal for applications that require exact anomaly identification. Meanwhile, Ensemble (1) achieves perfect recall despite a minor drop in precision, effectively catching all abnormalities in the dataset. The ensembles' overall excellent accuracy underlines their trustworthiness in anomaly detection applications.

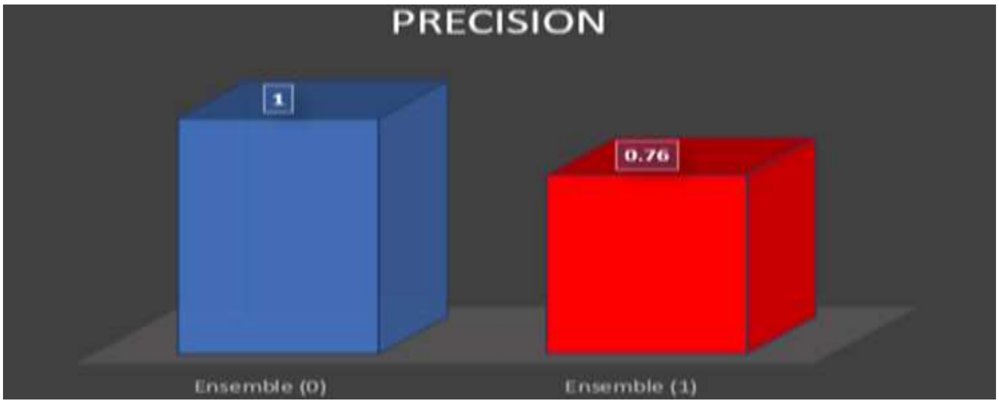

Figure 8: Precision of proposed machine-learning technique bar graph

Figure 8 shows that Ensemble (0) has a flawless precision score of 1.00, meaning that all detected anomalies are true positives with no false positives. Ensemble (1), on the other hand, has a precision of 0.76, implying that while it finds a wider range of abnormalities, some of them may be false positives. The great precision of Ensemble (0) demonstrates its capacity to correctly identify true abnormalities without misclassification.

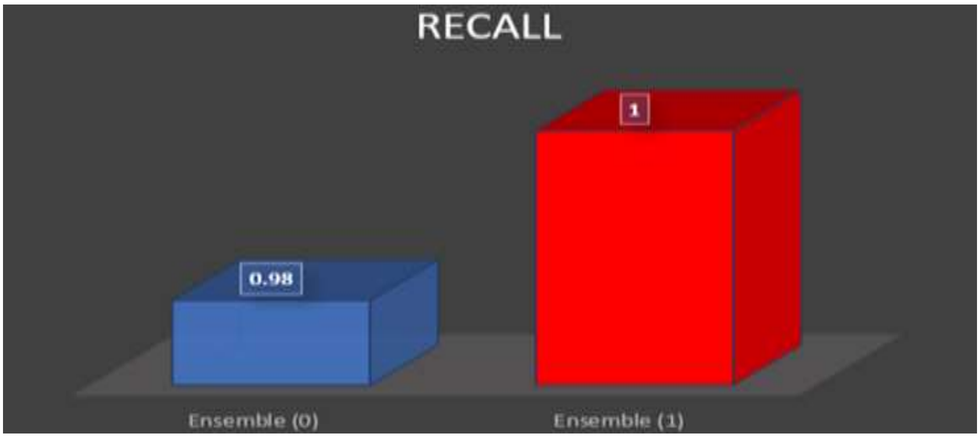

Figure 9: Recall of proposed machine-learning technique bar graph

Figure 9 shows that Ensemble (0) has a recall of 0.98, suggesting strong sensitivity in spotting abnormalities. Ensemble (1) outperforms this parameter, with a recall score of 1.00, indicating that it correctly detects all abnormalities in the dataset. Ensemble (1)'s faultless recall ensures that no anomaly is missed, making it a comprehensive anomaly detection tool.

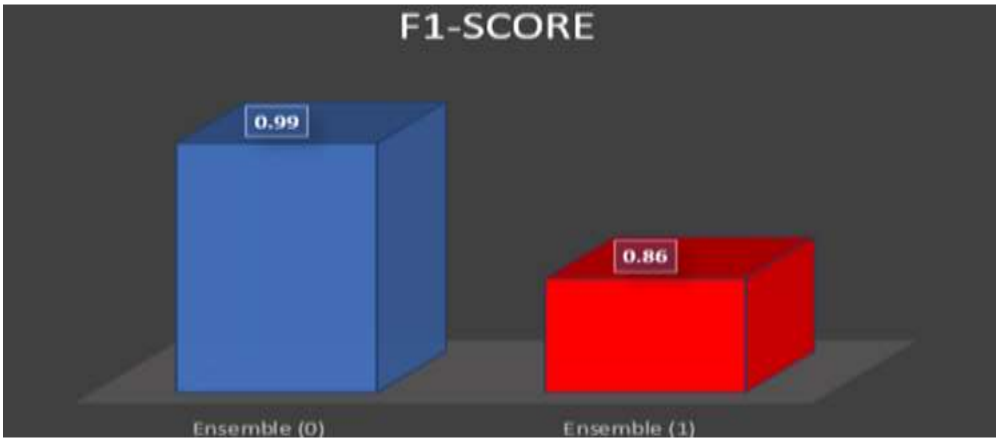

Figure 10: F1-Score of proposed machine-learning technique bar graph

Figure 10 shows that Ensemble (0) has an F1-Score of 0.99, suggesting superior overall performance. Ensemble (1) has an F1-Score of 0.86, which indicates balanced but slightly lower performance due to reduced precision. Ensemble (0)'s strong F1-Score demonstrates its ability to accurately recognize and classify anomalies.

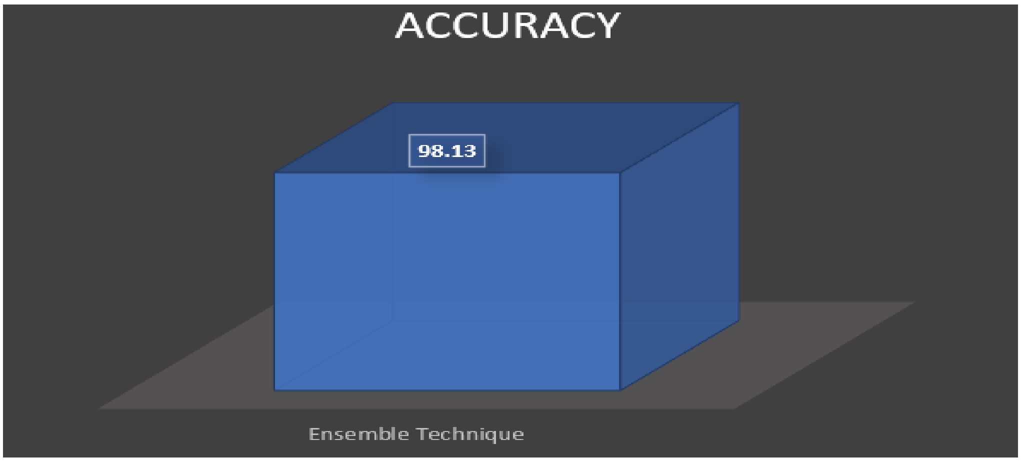

Figure 11: Accuracy of proposed machine-learning technique bar graph

Figure 11 shows that the accuracy of the improvised machine-learning technique is 98.13%, indicating that 98.13% of all cases (both normal and anomalous) are identified correctly. This high accuracy demonstrates the technique's dependability in ensuring comprehensive detection integrity. The 98.13% accuracy score verifies the technique's ability to make accurate predictions. Overall, the improvised technique exhibits effective anomaly detection capabilities while maintaining stable performance across critical parameters.

VIII. CONCLUSION

Machine learning techniques for Anomaly Detection are employed to learn from datasets, both normal and expected patterns and anomalies in the data. These algorithms can then identify new data points that deviate significantly from what is estimated, flagging them as possible anomalies. The anomalies have been spotted in the identified datasets, and the correlation matrix has been calculated. The three machine learning techniques for anomaly detection algorithms namely DBSCAN, Isolation Forest, and Autoencoder, are built for the vehicle insurance claim fraud detection dataset, and their accuracies are evaluated to determine the most efficient algorithm.

When great accuracy is desired, the Autoencoder model is the optimal choice. Its ability to capture complicated data patterns makes it very dependable for anomaly detection. On the other hand, DBSCAN is famous for its practical usefulness in clustering-based anomaly detection, particularly in complex datasets with varying densities. While both algorithms have advantages, the best one depends on the specific requirements of the anomaly detection task. If precision is critical, an autoencoder is the way to go. DBSCAN works better than other algorithms when dealing with dense, complex data and it detects anomalies more effectively. An improvised DBSCAN, Autoencoder, and Isolation Forest can be a useful method for detecting anomalies with an accuracy of 98.13%. This strategy has the potential to attain high precision, recall, and F1-score by utilizing their complementary strengths, which could result in enhanced anomaly identification when compared to using a single technique.

Future directions include investigating several forms of autoencoders (for example, Variational Autoencoders and Denoising Autoencoders) to increase anomaly detection robustness and accuracy. Experiment with deeper and more complicated autoencoder architectures to detect more intricate patterns in data.

IX. REFERENCES

[1] Arjunan, T., 2024. Fraud Detection in NoSQL Database Systems Using Advanced Machine Learning. International Journal of Innovative Science and Research Technology(IJISRT), March, 13, pp.248-53.

[2] Hansson, A. and Cedervall, H., 2022. Insurance Fraud Detection using Unsupervised Sequential Anomaly Detection.

[3] Reis, T., Kreibich, A., Bruchhaus, S., Krause, T., Freund, F., Bornschlegl, M.X. and Hemmje, M.L., 2022. An Information System Supporting Insurance Use Cases by Automated Anomaly Detection. Big Data and Cognitive Computing, 7(1), p.4.

[4] "Insurance Fraud," Federal Bureau of Investigation. Accessed: Jun. 7, 2024 at 6:30 AM. [Online]. Available: https://www.fbi.gov/stats-services/publications/insurance-fraud

[5] J. Liu et al., "Anomaly Detection in Manufacturing Systems Using Structured Neural Networks," in 2018 13th World Congress on Intelligent Control and Automation (WCICA), Changsha, China: IEEE, Jul. 2018, pp. 175–180. doi: 10.1109/WCICA.2018.8630692.

[6] Cooper, S., Bhuiyan, M. and Arslan, E., 2020, October. Machine learning for data transfer anomaly detection. In IEEE/ACM Supercomputing.

[7] V. Ceronmani Sharmila, K. K. R., S. R., S. D., and H. R., "Credit Card Fraud Detection Using Anomaly Techniques," in 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), CHENNAI, India: IEEE, Apr. 2019, pp. 1–6. doi: 10.1109/ICIICT1.2019.8741421.

[8] Bouman, R., Bukhsh, Z. and Heskes, T., 2024. Unsupervised anomaly detection algorithms on real-world data: how many do we need?. Journal of Machine Learning Research, 25(105), pp.1-34.

[9] "What Is Supervised Learning? | IBM." Accessed: Jun.12, 2024 at 5:17 PM. [Online]. Available: https:// www. ibm.com/topics/supervised-learning

[10] "(31) Semi-Supervised Learning and its Application | LinkedIn." Accessed: Jun. 5, 2024 at 3:12 PM. [Online]. Available: https://www.linkedin.com/pulse/semi-supervised-learning-its-application-crafsol-technology/

[11] "What Is Unsupervised Learning? | IBM." Accessed: Jun. 17, 2024 at 2:15 PM. [Online]. Available: https://www.ibm.com/topics/unsupervised-learning

[12] V. Yepmo, G. Smits, M.-J. Lesot, and O. Pivert, "Leveraging an Isolation Forest to Anomaly Detection and Data Clustering," Data Knowl. Eng., vol. 151, p. 102302, May 2024, doi: 10.1016/j.datak.2024.102302.

[13] Ataccama, "What is Anomaly Detection," Ataccama. Accessed: May 10, 2024 at 7:17 PM. [Online]. Available: https://www.ataccama.com/blog/what-is-anomaly-detection

[14] "(23) 'Exploring Anomaly Detection with Isolation Forest Algorithm: A Comprehensive Guide' | LinkedIn." Accessed: May 17, 2024. [Online]. Available: https://www.linkedin.com/pulse/exploring-anomaly-detection-isolation-forest-algorithm-mukhtar-shaikh/

[15] A. Toshniwal, K. Mahesh, and J. R., "Overview of Anomaly Detection techniques in Machine Learning," in 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India: IEEE, Oct. 2020, pp. 808–815. doi: 10.1109/I-SMAC49090.2020.9243329.

[16] "DBSCAN," Wikipedia. May 11, 2024. Accessed: May 17, 2024 at 6: 19 AM. [Online]. Available: https://en.wikipedia.org/w/index.php?title=DBSCAN&oldid=1223366932

[17] C.-W. Tien, T.-Y. Huang, P.-C. Chen, and J.-H. Wang, "Using Autoencoders for Anomaly Detection and Transfer Learning in IoT," Computers, vol. 10, no. 7, p. 88, Jul. 2021, doi: 10.3390/computers10070088.

[18] G. Bae, S. Jang, M. Kim, and I. Joe, "Autoencoder-Based on Anomaly Detection with Intrusion Scoring for Smart Factory Environments," in Parallel and Distributed Computing, Applications and Technologies, vol. 931, J. H. Park, H. Shen, Y. Sung, and H. Tian, Eds., in Communications in Computer and Information Science, vol. 931. , Singapore: Springer Singapore, 2019, pp. 414–423. doi: 10.1007/978-981-13-5907-1_44.

[19] M. Munir, M. A. Chattha, A. Dengel, and S. Ahmed, "A Comparative Analysis of Traditional and Deep Learning-Based Anomaly Detection Methods for Streaming Data," in 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA: IEEE, Dec. 2019, pp. 561–566. doi: 10.1109/ICMLA.2019.00105.

[20] "Anomaly detection," Wikipedia. Apr. 25, 2024. Accessed: May 10, 2024 at 6:18 PM. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Anomaly_detection&oldid=1220713426

[21] "Anomaly detection - Things Solver." Accessed: May 12, 2024 at 8:12 AM. [Online]. Available: https://thingsolver.com/blog/anomaly-detection/

[22] Xin, R., Liu, H., Chen, P. and Zhao, Z., 2023. Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework. Journal of Cloud Computing, 12(1), p.7.

[23] Bhadri Naarayanan P and Sri Venkateswara College of Engineering, "Comparing the Performance of Anomaly Detection Algorithms," Int. J. Eng. Res., vol. V9, no. 07, p. IJERTV9IS070532, Jul. 2020, doi: 10.17577/IJERTV9IS070532.

[24] A. Diro, N. Chilamkurti, V.-D. Nguyen, and W. Heyne, "A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms," Sensors, vol. 21, no. 24, p. 8320, 2021.

[25] Özkum, Ö., 2023. CREDIT CARD FRAUD DETECTION WITH AUTOENCODERS, ONE-CLASS SVMS AND ISOLATION FORESTS (Master's thesis, Middle East Technical University).

[26] M. Gupta Swati, S. Patel, S. Kumar, and G. Chauhan, "ANOMALY DETECTION IN CREDIT CARD TRANSACTIONS USING MACHINE LEARNING," Int. J. Innov. Res. Comput. Sci. Technol., vol. 8, no. 3, May 2020, doi: 10.21276/ijircst.2020.8.3.5.

[27] D. S. Maurya, Y. A. Balushi, and J. Kharade, "Integrity Shield: Ensuring Real-time Data Integrity in Healthcare IoT with Isolation Forest Anomaly Detection," Int. J. Intell. Syst. Appl. Eng..

[28] S. H. Haji and S. Y. Ameen, "Attack and Anomaly Detection in IoT Networks using Machine Learning Techniques: A Review," Asian J. Res. Comput. Sci., pp. 30–46, Jun. 2021, doi: 10. 9734/ ajrcos/ 2021/ v 9i 230218.

[29] I. Petrariu, A. Moscaliuc, C. E. Turcu, and O. Gherman, "A Comparative Study of Unsupervised Anomaly Detection Algorithms used in a Small and Medium-Sized Enterprise," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 9, 2022, doi: 10.14569/IJACSA.2022.01309108.

[30] C. Das, A. Rasool, A. Dubey, and N. Khare, "Analyzing the Performance of Anomaly Detection Algorithms," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120649.

[31] WyWiol, T.L., 2023. Condition Monitoring Of Machine Components From Drive Data Using Semi-Supervised Anomaly Detection Methods.

[32] K. Demestichas, N. Peppes, T. Alexakis, and E. Adamopoulou, "An Advanced Abnormal Behavior Detection Engine Embedding Autoencoders for the Investigation of Financial Transactions," Information, vol. 12, no. 1, p. 34, Jan. 2021, doi: 10.3390/info12010034.

[33] Abdullah, A. and Arjunan, T., 2023. Leveraging Advanced Machine Learning Techniques for Enhanced Intrusion and Fraud Detection in NoSQL Database Systems. International Journal of Applied Machine Learning and Computational Intelligence, 13(11).

[34] M. Da Silva Ferreira, L. F. Vismari, P. S. Cugnasca, J. R. De Almeida, J. B. Camargo, and G. Kallemback, "A Comparative Analysis of Unsupervised Learning Techniques for Anomaly Detection in Railway Systems," in 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA: IEEE, Dec. 2019, pp. 444–449. doi: 10.1109/ICMLA.2019.00083.

[35] V. Wilmet, S. Verma, T. Redl, H. Sandaker, and Z. Li, "A Comparison of Supervised and Unsupervised Deep Learning Methods for Anomaly Detection in Images." arXiv, Jul. 19, 2021. Accessed: May 10, 2024. [Online]. Available: http://arxiv.org/abs/2107.09204

[36] R. Xin, H. Liu, P. Chen, P. Grosso, and Z. Zhao, "FIRED: a fine-grained robust performance diagnosis framework for cloud applications," Future Gener. Comput. Syst., vol. 155, pp. 300–311, Jun. 2024, doi: 10.1016/j.future.2024.02.014.

[37] C. Chahla, H. Snoussi, L. Merghem, and M. Esseghir, "A deep learning approach for anomaly detection and prediction in power consumption data," Energy Effic., vol. 13, no. 8, pp. 1633–1651, Dec. 2020, doi: 10.1007/s12053-020-09884-2.

[38] Zangrando, N., Fraternali, P., Petri, M., Pinciroli Vago, N.O. and Herrera González, S.L., 2022. Anomaly detection in quasi-periodic energy consumption data series: a comparison of algorithms. Energy Informatics, 5(Suppl 4), p.62.

[39] Corizzo, R., Ceci, M., Pio, G., Mignone, P. and Japkowicz, N., 2021, October. Spatially-aware autoencoders for detecting contextual anomalies in geo-distributed data. In International conference on discovery science (pp. 461-471). Cham: Springer International Publishing.

[40] W. Fernandes, K. S. Komati, and K. Assis De Souza Gazolli, "Anomaly detection in oil-producing wells: a comparative study of one-class classifiers in a multivariate time series dataset," J. Pet. Explor. Prod. Technol., vol. 14, no. 1, pp. 343–363, Jan. 2024, doi: 10.1007/s13202-023-01710-6.

[41] Rezapour, M., 2019. Anomaly detection using unsupervised methods: credit card fraud case study. International Journal of Advanced Computer Science and Applications, 10(11).

[42] F. Falcão et al., "Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection," in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol Cyprus: ACM, Apr. 2019, pp. 318–327. doi: 10.1145/3297280.3297314.

[43] G. Princz, M. Shaloo, and S. Erol, "Anomaly Detection in Binary Time Series Data: An unsupervised Machine Learning Approach for Condition Monitoring," Procedia Comput. Sci., vol. 232, pp. 1065–1078, 2024, doi: 10.1016/j.procs.2024.01.105.

[44] Schindler, T.F., Schlicht, S. and Thoben, K.D., 2023. Towards Benchmarking for Evaluating Machine Learning Methods in Detecting Outliers in Process Datasets. Computers, 12(12), p.253.

[45] Kerstinga, M.C., Patrikara, A.M., Schneidera, E., Kusunoki-Martina, T., Drummb, A. and O'Neillb, C., Condition-Based Maintenance Using Unsupervised Time-Series Anomaly Detection.

[46] Çelik, M., Dadaşer-Çelik, F. and Dokuz, A.Ş., 2011, June. Anomaly detection in temperature data using DBSCAN algorithm. In 2011 international symposium on innovations in intelligent systems and applications (pp. 91-95). IEEE.

[47] Sheridan, K., Puranik, T.G., Mangortey, E., Pinon-Fischer, O.J., Kirby, M. and Mavris, D.N., 2020. An application of dbscan clustering for flight anomaly detection during the approach phase. In AIAA Scitech 2020 Forum (p. 1851).

[48] "Find Open Datasets and Machine Learning Projects | Kaggle." Accessed: Jun. 17, 2024 at 5:17 PM. [Online]. Available: https://www.kaggle.com/datasets

[49] R. Jodha, "One Hot encoding," Scaler Topics. Accessed: Jun. 12, 2024 at 7:34 AM. [Online]. Available: https://www.scaler.com/topics/data-science/one-hot-encoding/

[50] R. Polikar, "Ensemble Learning," in Ensemble Machine Learning: Methods and Applications, C. Zhang and Y. Ma, Eds., New York, NY: Springer, 2012, pp. 1–34. doi: 10.1007/978-1-4419-9326-7_1.

[51] "Ensemble learning," Wikipedia. May 20, 2024. Accessed: Jun. 16, 2024 at 9:19 PM. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Ensemble_learning&oldid=1224773391

[52] "(31) Confusion Matrix, Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures | LinkedIn." Accessed: Jun. 27, 2024 at 10: 12 AM [Online]. Available: https://www.linkedin.com/pulse/confusion-matrix-accuracy-precision-recall-f1-score-measures-silwal/