Arima

ARIMA is an acronym that stands for Auto-Regressive Integrated Moving Average. The ARIMA model is a well-liked and commonly applied statistical technique for time series forecasting. The two most popular methods for predicting time series are exponential smoothing and ARIMA models, both of which offer complimentary approaches to the issue. While ARIMA models seek to capture the autocorrelations in the data, exponential smoothing methods are based on a description of the trend and seasonality in the data.It belongs to a family of models that may represent a variety of common temporal patterns in time series data.A family of statistical models for assessing and predicting time series data is known as an ARIMA model. Although it is really simple to use, this approach is quite effective.

Advantages

- ARIMA is a reasonably simple and easy-to-understand model, making it accessible to a wider audience that do not have substantial mathematical experience.
- Interpretability: It is possible to acquire insights into the underlying dynamics and patterns of the time series by interpreting the model's AR, I, and MA components.
- Absence of external variables: ARIMA can be used with univariate time series data in situations when alternative models may not be acceptable or accessible due to the lack of other predictors or characteristics.
- Effective with little datasets: ARIMA may perform well even with tiny quantities of data, making it ideal for situations where data gathering may be restricted.
- ARIMA captures autocorrelation and delayed dependencies in time series data, making it suited for datasets having temporal patterns

Disadvantages

- Limitations when dealing with complicated interactions: Because ARIMA assumes linear relationships, it might not be able to fully capture more complex and nonlinear patterns in the data.
- Sensitive to outliers: The ARIMA model's sensitivity to outliers may have an effect on how well it predicts the future.
- Manual parameter tuning: Choosing the right order of the ARIMA model (p, d, q) frequently takes trial and error, making it a bit laborious and time-consuming.
- Limited long-term projections: ARIMA may perform poorly when projecting over a long horizon because it may be unable to capture complicated patterns and seasonality outside of the model's order.
- Lack of handling of seasonality: ARIMA struggles to handle time series data with numerous seasonal patterns or irregular frequencies.

One class- SVM

One-Class SVM (Support Vector Machine) is a popular algorithm used for anomaly detection, particularly when dealing with unimodal data (data with only one class of normal instances). It is well-suited for scenarios where labeled anomalies are scarce or unavailable. Here's how the One-Class SVM works in anomaly detection

Training phase:

- One-Class SVM is trained on a dataset containing only normal instances, which is the "one class" it aims to learn.
- The algorithm finds the hyperplane that best separates the normal instances from the origin (or center) of the feature space, while minimizing the number of support vectors (data points closest to the decision boundary).

Anomaly detection phase:

- Once the One-Class SVM is trained, it can be used to identify anomalies in new, unseen data points.
- During inference, data points that are located far from the learned hyperplane are considered potential anomalies, as they deviate significantly from the majority of normal instances in the feature space.

Advantages

- Effective with limited anomaly data: One-Class SVM can perform well even with a small number of labeled anomalies, making it suitable for scenarios with scarce anomaly samples.
- Robust to high-dimensional data: The algorithm is capable of handling high-dimensional feature spaces efficiently, which is beneficial when dealing with complex data.
- Nonlinear kernel support: One-Class SVM can handle nonlinear relationships in the data by using a kernel trick, such as the Radial Basis Function (RBF) kernel.
- Good for unimodal data: When the majority of the data belongs to a single class (normal class) and the anomaly class differs significantly, One-Class SVM can be effective in detecting anomalies.

Disadvantages

- Sensitivity to hyperparameters: One-Class SVM relies on parameters like the kernel, regularization, and nu (a parameter controlling the trade-off between maximizing the margin and allowing misclassifications). Selecting appropriate hyperparameters may require careful tuning.
- Challenging for multiclass data: One-Class SVM is not suitable for scenarios where data contains multiple classes, as it can only learn one class during training.

- Outlier contamination: If the training data contains a significant number of outliers or anomalies, the One-Class SVM might not be able to generalize well to detect unseen anomalies.
- Scalability: One-Class SVM may become computationally expensive on large datasets, particularly when using nonlinear kernels.

Auto-encoders

An effective neural network design for anomaly detection is autoencoders. When there is a lack of labeled anomaly data or it is unavailable, they are especially well suited for unsupervised anomaly detection jobs. In order to attempt to reconstruct the original data, autoencoders first learn a compressed representation of the input data in a lower-dimensional space. Based on how successfully the autoencoder can recreate the input data, anomalies are found. To efficiently utilize autoencoders for anomaly detection, following steps are involved:

the training stage

- An exclusively normal dataset (data devoid of abnormalities) is used to train the autoencoder.
- The architecture comprises of a decoder that reconstructs the data from the latent space and an encoder that compresses the input data into a lower-dimensional representation (latent space).
- The reconstruction error—the difference between the input data and the reconstructed output—is something the autoencoder learns to reduce during training. Making the reconstructed data as similar to the original input as feasible is the goal.

anomaly detection phase:

- After the autoencoder has been trained on typical data, it may be used to spot anomalies in fresh, unused data.
- Data points that drastically depart from the taught patterns are flagged during inference.Higher reconstruction mistakes will be present in the training data.
- The reconstruction error is given a threshold, and data points with reconstruction errors greater than the threshold are treated as possible anomalies.

Advantages

- Autoencoders are useful for situations where getting labeled anomalies may be challenging or expensive since they do not require labeled anomaly data for training.
- Nonlinearity: Autoencoders are particularly beneficial for identifying anomalies in high-dimensional and complicated datasets because they can efficiently capture complex nonlinear correlations in the data.
- Autoencoders spontaneously learn a compressed representation of the data, which reduces the dimensionality of the feature space and can help with data visualization and understanding.
- Flexibility: Autoencoders are adaptable for diverse anomaly detection jobs since they can be tailored and adjusted to various sorts of data.

Disadvantages

- Overfitting: When the model is too sophisticated or the training dataset is small, autoencoders may overfit to the training data.
- Choosing a good threshold for the reconstruction error can be difficult, and for maximum performance, it may be necessary to carefully tune and validate the threshold.
- Limited interpretability: Although autoencoders are good at identifying patterns, it may be more difficult to interpret the learnt latent space than it is with conventional statistical models.

K-nearest Neighbour

KNN is a particular kind of supervised learning algorithm used for both regression and classification applications. A new data point in classification is given a class label based on the dominant class among its k-nearest neighbors by the algorithm. By averaging (or weighting) the target values of its k-nearest neighbors, regression predicts a continuous value. The KNN algorithm is regarded as a non-parametric approach since it does not rely on any presumptions regarding the distribution of the underlying data. To create predictions when presented with new data, it memorizes the complete training dataset.

A crucial hyperparameter in KNN is the choosing of the value of k (the number of nearest neighbors). While a big k number may result in excessively smooth decision limits and the danger of overlooking local patterns, a little k value may produce noisy and unstable forecasts.There are other distance metrics besides Euclidean distance that may be used to gauge how similar two data points are. Despite the fact that Euclidean distance is frequently utilized, alternative distance metrics, such as Manhattan distance or cosine similarity, can also be used depending on the data and the issue at hand.

Overall, KNN is a straightforward and intuitive approach that can be useful for some sorts of situations, particularly when the decision boundary is nonlinear and complicated. Due to the need to save and compute distances to every training data points during prediction, which may be computationally costly for big datasets, it may not be scalable. However, KNN may be a useful tool for classification and regression problems with the right parameter tweaking and consideration of the distance metric.

<u>advantages</u>

- Simple and intuitive: KNN is uncomplicated and simple to comprehend. Because it doesn't need complicated mathematical formulae, amateurs and non-experts may use it.
- No Training step: Because KNN is a lazy learner algorithm, the training step does not include explicit model learning. Instead, it memorizes the training data, accelerating and optimizing memory use during training.
- Non-parametric: KNN makes no assumptions regarding the distribution of the underlying data. It can handle connections between characteristics and the target variables that are complicated and nonlinear.
- Flexibility: KNN is adaptable to a variety of issues since it can be used for both classification and regression tasks.

<u>Disadvantages</u>

- Costly to Calculate: When making a prediction, KNN must determine how far each new data point is from each training data point. This can be a costly process when dealing with huge datasets.
- Sensitivity to Irrelevant characteristics: KNN is sensitive to irrelevant characteristics since it treats all features equally when determining distances. In order to assure the algorithm's effectiveness, preprocessing and feature selection become critical.

- Correct Scaling is Required: KNN is sensitive to the scale of the features. Large-scale features may predominate in the distance computation, producing skewed results. It is required to scale the features to get around this problem.
- Finding the Value of k: It's important to choose the right value for k. A low k number might produce noisy forecasts, whereas a high k value could produce overly smooth decision limits.

CBLOF

Cluster-based Local Outlier Factor (CBLOF) is an anomaly detection algorithm that combines the concepts of Local Outlier Factor (LOF) with clustering techniques. It is designed to identify local outliers, which are data points that have significantly different characteristics from their neighboring points in the dataset.

Here is how CBLOF functions:

- CBLOF begins by grouping the data points into clusters based on how similar they are. It groups data points that are adjacent to one another in the feature space using a clustering method (like k-means, for example).
- LOF Calculation: CBLOF determines the Local Outlier Factor (LOF) for each data point when clustering is complete. The local density deviation of a data point in relation to its neighbors is measured by the LOF. It measures the difference in density between a data point's neighbors.
- Outlier Score: Each data point's outlier score is then calculated using the LOF values. The outlier score reveals how far a data point deviates from the norm in relation to its immediate surroundings. A data point is more likely to be an outlier if its outlier score is higher.
- Thresholding: Using the outlier scores, CBLOF establishes a threshold value. Anomalies or outliers are defined as data points having outlier scores that exceed the threshold.

Advantages

- Effective for Detecting Local Outliers: CBLOF is very helpful for finding local outliers that might exist in various parts of the dataset with varied densities.
- Adaptable to Different Cluster Shapes: By using clustering, CBLOF is better able to manage non-spherical or unevenly formed clusters.
- Parameter-Free: CBLOF is reasonably straightforward to use, especially for users who want an anomaly detection approach without a lot of manual parameter tuning. It has few hyperparameters to modify.

- Interpretable: The CBLOF outlier scores may be read as a measure of outlierness in relation to the immediate neighborhood, making the findings simple to comprehend and analyze.

Disadvantages

- Sensitivity to Number of Clusters: The number of clusters utilized in the initial clustering process can have an impact on how well CBLOF performs. The detection of outliers could be impacted if the number of clusters is not accurately approximated.

- Scalability: Because CBLOF involves clustering and LOF computations for each data point, its computational cost might be an issue for big datasets or high-dimensional data.

- Limitations for Global Outliers: CBLOF is primarily developed for locating local outliers. In identifying global outliers that span numerous clusters, it might not perform as well.

- Initialization Sensitivity: The clustering algorithm's initialization can have an impact on how well CBLOF performs, and various initializations may provide different outcomes.

Isolation-Forest

Isolation Forest is another common anomaly detection technique that is built on the notion of separating abnormalities in a forest of trees. In 2008, Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou published a paper named "Isolation Forest" in which they first described it. For finding abnormalities in high-dimensional data, the approach is effective and efficient.

Here's how the isolation forest works:

- Random Subsampling: The algorithm begins by picking at random a subset of the dataset's data points. One isolation tree will be built from this subset as its foundation.
- Each isolation tree is built iteratively in recursive partitioning. The data points are divided along the value of a randomly selected attribute at each stage. Up until the data points are isolated, they are divided repeatedly, resulting in each data point becoming a leaf node of the tree.

- route Length: In the isolation tree, the distance along the route from the root to the leaf node is used to gauge how isolated a data point is. Since anomalies are simpler to isolate than regular data points, it is predicted that they will have shorter route lengths.
- Ensemble of Trees: Using recursive partitioning and random subsampling, several isolation trees are produced. The group of trees works together to detect abnormalities in the dataset.
- Anomaly Score: The average path length from the ensemble of trees is computed for each data point. A shorter average path length denotes a higher possibility of becoming an anomaly, and it is used as an anomaly score.
- Thresholding: To identify which data points are anomalies, a threshold is utilized. Anomalies are defined as data points with average path lengths below the threshold whereas data points with average path lengths above the threshold are deemed normal.

Advantages

- Scalability: Isolation Forest is suited for real-world applications because it is computationally efficient, handles big datasets, and handles high-dimensional data efficiently.
- No Assumptions: The technique is successful at handling data with complex patterns and makes no assumptions about the distribution of the underlying data.
- Low Computational Cost: When compared to other anomaly detection approaches like density-based ones, Isolation Forest has a lower computational cost.
- Isolation Forest is proficient in identifying outliers in several dimensions, which is particularly helpful for high-dimensional data.

Disadvantages

- Sensitivity to Number of Trees: The quantity of trees in the ensemble can affect how well Isolation Forest performs. The accuracy of anomaly detection may be impacted by the selection of the wrong number of trees.
- Choosing a suitable threshold to categorize anomalies can be difficult and may need for thorough adjustment and confirmation.
- Limited to Unimodal Data: Isolation Forest is primarily intended for isolating single-mode anomalies, hence it may not perform as well with multimodal data distributions.

Random forest

The widely used ensemble learning technique Random Forest is utilized for both classification and regression applications. It may be utilized as a basis model inside an anomaly detection framework even if it is not explicitly made for anomaly detection. One such method is the Isolation Forest, which utilizes isolation trees and the Random Forest idea to find anomalies.

Here is an example of how to apply Random Forest to the Isolation Forest for anomaly detection:

- Isolation Trees: The Isolation Forest employs isolation trees rather than the standard decision trees used in Random Forest. These isolation trees are modified decision trees that are built in a way that more effectively isolate anomalies.
- Isolation Forest randomly subsamples a subset of data points in a manner similar to Random Forest in order to create an isolation tree.
- Recursive Partitioning: Based on randomly chosen characteristics and splitting values, the data points are recursively partitioned to create the isolation trees.
- Path Length: A data point's depth in an isolation tree, or the number of splits necessary to isolate it, is used to gauge how isolated it is.
- The average path length from the ensemble of isolation trees is determined for each data point to get the anomaly score. The data point is easier to identify and is more likely to be an anomaly if the average journey length is lower.
- Thresholding: To identify which data points are anomalies, a threshold is utilized. Anomalies are defined as data points with average path lengths below the threshold whereas data points with average path lengths above the threshold are deemed normal.

In conclusion, Random Forest may be used successfully in the context of the Isolation Forest to discover anomalies even if it is not a method for anomaly detection in and of itself. In high-dimensional datasets with complicated patterns, the combination of Random Forest and isolation trees enables rapid and reliable anomaly identification. To get the best results, though, thorough parameter adjustment and threshold selection are crucial.

<u>Advantages</u>

- Ensemble Effect: When compared to utilizing a single isolation tree, employing numerous isolation trees in an ensemble enables more reliable and accurate anomaly identification.
- Scalability: Random Forest and the Isolation Forest are capable of handling enormous datasets and high-dimensional data because to their computational efficiency.
- Robustness: Random Forest prevents false positives in anomaly detection since it is less prone to overfitting than individual decision trees.
- Flexibility: Random Forest is appropriate for a variety of data sources since it can handle both category and numerical characteristics.

<u>Disadvantages</u>

- Selecting an acceptable threshold to categorize anomalies may be difficult and may need for thorough adjustment and validation, similar to how the Isolation Forest was used.
- Global Outliers: Random Forest and the Isolation Forest may not be as effective in spotting outliers that cut across numerous clusters or geographic areas.

<u>ABOD</u>

An anomaly detection system called ABOD (Angle-Based Outlier Detection) uses the angles created between data points in a high-dimensional space to identify outliers. In their study titled "Angle-Based Outlier Detection in High-dimensional Data" from 2008, Kriegel et al. introduced it. Due to the dimensionality curse, standard distance-based approaches may not be as successful in high-dimensional data environments as ABOD.

Here is how ABOD functions:

- Pairwise Distances: ABOD begins by figuring out how far apart each data point in the dataset is from each other. In this stage, the distance or dissimilarity between each pair of data points is calculated (for example, using the Euclidean distance).

- Angle Calculation: ABOD calculates the angles generated between the pairwise distance vectors for each data point by taking into account its neighbors, or other data points in the dataset. The angles are calculated as the angle between the two vectors coming from the subject data point and linking it to two separate neighbors.

- Outlier Score: Based on the variation of the angles made with its neighbors, each data point's outlier score is calculated. Because they differ greatly from their neighbors, outliers have larger angular variance than typical data points.

- Data points are classified as anomalies or normal points using a threshold. Anomalies are data points having outlier scores greater than the threshold.

Advantages

- Effective with High-Dimensional Data: Due to the "curse of dimensionality," standard distance-based algorithms may not perform as well in high-dimensional data. However, ABOD is intended to handle high-dimensional data successfully.
- Insensitive to Scale: ABOD is ideal for datasets with various feature scales since it is not sensitive to the scale of the features.
- Robustness: Compared to certain other distance-based anomaly detection techniques, ABOD is less impacted by noise and outliers.
- Computing Effectiveness: ABOD is appropriate for huge datasets because of its moderate computational complexity.

Disadvantages

- Sensitivity to Noise: ABOD can still be vulnerable to outliers and noisy data points, while being more resistant to noise than some other techniques.
- Limitations in Detecting Global Outliers: ABOD is primarily meant to find local outliers. In locating global outliers that traverse numerous clusters or regions, it might not perform as well.
- Selecting a suitable threshold for identifying anomalies may be difficult and may require extensive adjustment and validation, as is the case with many anomaly detection methods.

In conclusion, the ABOD technique is useful for locating local outliers in high-dimensional datasets. It is a handy tool for anomaly detection jobs in situations where conventional distance-based approaches might not be as successful due to its efficacy and processing efficiency. However, it might not be able to detect global outliers, and careful threshold selection is required to get accurate and dependable anomaly detection findings.

Support Vector Machine

A strong machine learning approach that may be utilized for anomaly detection is called Support Vector Machines (SVM). SVM is frequently used in supervised learning tasks like classification and regression, but it may also be used in unsupervised circumstances like anomaly detection.

SVM functions as follows in the context of anomaly detection:

- One-Class SVM: The One-Class SVM is the most used SVM variation for anomaly identification. Assuming that anomalies are uncommon and are not represented in the training data, it is trained on a dataset that only contains normal data. The One-Class SVM's objective is to discover a boundary around the typical data points in a high-dimensional feature space.
- Boundary Separation: The One-Class SVM seeks to identify the hyperplane that maximizes the margin around the typical data points while encoding the fewest number of outliers. The region contained by the decision boundary, represented by the hyperplane, is referred to as the "normal" region.
- Support Vector Machines (SVM) are a powerful machine learning method that may be used for anomaly identification. SVM may be utilized in unsupervised situations, such as anomaly detection, in addition to supervised learning tasks like classification and regression.

  Advantages

- SVM works effectively in high-dimensional feature spaces, making it appropriate for anomaly identification in complicated and high-dimensional data. Effective in High-Dimensional Spaces.
- SVM has a wide range of applications, including binary and multi-class anomaly detection.
- Robustness: SVM can be useful when working with noisy datasets since it is less susceptible to outliers in the training data.
- Ability to Handle Nonlinear Data: By utilizing kernel functions to map data into higher-dimensional spaces, SVM is able to handle nonlinear connections.

  Disadvantages

- anomaly Ratio: The data's anomaly ratio has a significant impact on how well SVM detects anomalies. The model may have trouble telling the difference between normal and anomalous data properly if the fraction of anomalies is too large.
- SVM needs thorough parameter adjustment in order to provide the best results. The performance of the model can be considerably influenced by the choice of the regularization parameter (C) and kernel function.
- SVM is a computationally intensive algorithm, especially when working with huge datasets or high-dimensional data.

- unbalanced Data: SVM may need extra strategies (e.g., resampling, altering class weights) to manage the class imbalance efficiently in cases when the data is severely unbalanced and anomalies are represented by a substantially smaller number of data points than usual.

SVM, particularly the One-Class SVM form, can, in conclusion, be a potent tool for anomaly identification, especially when working with high-dimensional and complicated data. The capacity to handle nonlinear data and resilience to outliers are also features it offers. To attain the optimum performance, however, rigorous parameter tweaking and attention to data imbalance are required.

LOF

An unsupervised anomaly detection approach called LOF (Local Outlier Factor) assesses the degree of outlierness of data points based on their local density in relation to their neighbors. In their work titled "LOF: Identifying Density-Based Local Outliers" published in 2000, Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander first proposed the concept.

LOF functions as follows:

- Calculation of Local Density: For each data point, LOF determines its local density by taking into account the separation from its k-nearest neighbors. When a data point is located in an area with numerous neighboring points, it is said to have a greater local density, indicating that it is more likely to be a part of the normal data.
- To determine how approachable a data point is from its k-nearest neighbors, LOF calculates the local reachability distance for each data point. This measurement of a data point's separation from its neighbors aids in the detection of outliers.
- The average ratio of a data point's local reachability distance to that of its k-nearest neighbors is used to calculate the local outlier factor (LOF). Outliers are data points with a LOF greater than 1, since they have a much greater local reachability distance than their neighbors.

<u>Advantages</u>

- Sensitivity to Local Context: LOF considers the local density of data points, enabling it to identify outliers accurately even in areas with varied data densities.
- Flexibility: LOF can handle complicated and non-linear patterns in the data and is not constrained to any particular data distribution.
- No Assumptions: LOF is a flexible technique for a variety of datasets since it makes no assumptions about the distribution of the underlying data or the number of clusters.
- Parameter-Free: LOF just needs to adjust one key hyperparameter (k, the number of neighbors), which minimizes the requirement for manual intervention.

<u>Disadvantages</u>

- Calculating the distances between each data point and its k-nearest neighbors is necessary for LOF, which can be computationally costly for big datasets.

- Sensitivity to k: The findings of the LOF are greatly impacted by the choice of the parameter k. If k is set incorrectly, the algorithm's performance may be affected.

- Scalability: For really big datasets, LOF's computational complexity may become a barrier.

- Cases that are on the boundary between being outliers or inliers may be difficult for LOF to identify.

In conclusion, LOF is a strong and adaptable method for anomaly identification, especially when working with datasets that have complicated patterns and varied data densities. It is good in locating local outliers without making any assumptions about the distribution of the data. To get the greatest performance with LOF, however, rigorous parameter tweaking and computational resource management are required.

PCA

Though Principal Component Analysis (PCA) is not specifically designed to aid in anomaly identification, it may be utilized as a preprocessing step or feature extraction method. PCA is a technique for dimensionality reduction that converts highly dimensional data into a lower-dimensional space while preserving the most crucial facts. By lowering the dimensionality, PCA can assist in visualizing and comprehending the data, which may help in spotting anomalies or patterns that are not immediately apparent in the original high-dimensional feature space.

Ways PCA may be applied to anomaly detection is as follows:

- Dimensionality Reduction: PCA creates a new set of orthogonal dimensions called principle components from the high-dimensional data that was initially there. These primary components are arranged so that the first component accounts for the majority of the variation in the data, the second component accounts for the same amount of variance, and so on.
- Retaining Important Information: The total explained variance is used to calculate the number of principle components to be retained. One can lower the dimensionality while keeping the bulk of the crucial information in the data by selecting a certain percentage of the cumulative explained variance (for example, 95% or 99%).
- Anomaly Detection in Reduced Space: After applying PCA to reduce the dimensionality, anomaly detection techniques may be used to search for anomalies in the lower-dimensional data. After PCA, certain popular approaches for detecting anomalies may be applied, such as clustering-based techniques like k-means, density-based techniques like DBSCAN, or distance-based techniques like k-Nearest Neighbors.

Advantages

- PCA can assist with the visualization of high-dimensional data in a lower-dimensional environment, making it simpler to see patterns or clusters that might signify abnormalities.

- Reduced calculation: PCA can result in quicker calculation and lower memory needs for future anomaly detection algorithms by lowering the dimensionality of the data.

- Feature Extraction: By concentrating on the most pertinent data, anomaly detection techniques may perform better thanks to PCA's ability to extract the most important features from the data.

Disadvantages

- Information Loss: Dimensionality reduction used in PCA may cause some information from the original data to be lost. Important details for anomaly identification may occasionally be overlooked.

- Preprocessing Overhead: Using PCA as a preprocessing step increases the pipeline's computing workload.

- Interpretability: It may be difficult to directly read the main components derived via PCA in the original feature space, which makes it more difficult to understand the findings of the anomaly detection.

In conclusion, when working with high-dimensional data, PCA can be a beneficial technique for anomaly identification. By making the data less dimensional, it may be easier to visualize and comprehend the data as well as perhaps enhance the effectiveness of following anomaly detection algorithms. When employing PCA in the anomaly detection process, however, the loss of information and interpretability should be taken into account.