Anomaly Detection: How to find rare/unusual instances/groups

Amrit Bhaskar
Computer Science and Engineering Program Department
Ira. A Fulton School of Engineering
Arizona State University
Tempe, United States
abhask10@asu.edu

***Abstract*—The project completed in the subject CSE 575 Statistical Machine Learning, Group Project, are summarized and highlighted in this report. The task is to detect anomalies or outliers in supervised and unsupervised data using different Machine Learning(ML) algorithms. This document aims to guide the readers to better understand the popular ML techniques to detect the anomalies. I'm working on this project for my master's degree in computer science portfolio.**

***Index Terms*—Anomaly Detection, Supervised, Unsupervised, Time series, Machine Learning**

I. Introduction

This document proposes different supervised and unsupervised solutions that exist to solve the anomaly detection problem and compare the results. A significant problem that has been researched for decades is finding anomalies. A variety of unique techniques have been created and are utilized to detect anomalies for various applications [1]. The difficulty of discovering patterns in data that do not conform to predicted behavior is known as anomaly detection. [2], [3]. A wide range of applications uses anomaly detection extensively. Examples of this include loan application processing, monitoring of medical conditions, and fraud detection [4]. Cyber security intrusion detection, defect detection for aviation safety study, streaming, and hyperspectral photography, among others, are other frequently utilized applications for detecting anomalies. The likelihood that unprotected data may include substantial, crucial, and actionable information raises the importance of identifying abnormalities across a variety of application domains. For instance, identifying an unusual pattern of computer network traffic could reveal an attack coming from a compromised system [5]. Another instance would be the identification of irregularities in the credit card transaction data that might be signs of theft [6]. Additionally, identifying an abnormality from an aviation sensor may reveal a failure in one or more aircraft components.

Arizona State University

Anomaly detection in time series data is an important application as there can exist a variety of anomalies like time series, regression anomalies, quantile anomalies, etc. Research on this topic is the aim of this project. Anomaly detection plays an important role in reliability when dealing with crucial data. When dealing with crucial data, outliers and anomalies can have a substantial negative influence. Anomalies can be detected using various data visualization and exploratory data analysis techniques. Unsupervised machine learning algorithms for detecting anomalies include Isolation Forest, Local Outlier Factor, Robust Covariance, One-Class SVM, SGD, Time series models, and others. If the target labels are present, supervised classification algorithms could be used. In the current project, we consider time-series numeric data via a sample dataset to classify data points into anomaly or not anomaly. We proceed to perform classification using XGBClassifier and improve the model performance using a better-performing model based on the F1 performance metrics.

Anomalies can occur in a wide range of business scenarios, including banking fraud detection, the failure rate of fintech transactions, and so on. To avoid too many false positives during the anomaly discovery process, proper anomaly detection should be able to distinguish signal from noise. Supervised learning is a scenario in which the model is trained on labeled data and then used to predict unlabeled data. In contrast, no labels are presented for data to train on in unsupervised learning. Each methodology has benefits and drawbacks. For example, supervised learning models produce highly accurate results, whereas unsupervised learning models do not improve performance over time. In our current project, we compare, analyze, and contrast various supervised and unsupervised models for anomaly detection using labeled and unlabeled data in order to determine which algorithms are efficient across all the time series.

We used 3 datasets comprising 2 labeled and 1 unlabelled dataset. The labeled datasets are - Retailer's price monitoring dataset [20] and Annthyroid dataset [21]. The unlabelled one

is Walmart's sales Dataset [19].

A. Retailer's price monitoring dataset

The data consists of 15,830 rows. It has 4 columns namely - timestamp, value, predicted, and is_anomaly. is_anomaly is the target label comprising of two unique values - 0 and 1.

B. Annthyroid dataset

The data consists of 7,200 rows. It has 6 real attributes and target label. The target data comprises of two unique values - 0 and 1.

C. Walmart's sales Dataset

The data consists of 421,570 rows. It's between the period 5th February 2010 to 1st November 2012. The data is stored at a week's frequency. The data is unlabelled and consists of columns namely - store number, department, date, weekly sales, holiday flag, temperature, fuel price, and a few promotional markdown features.

II. Methodology

We had access to both Labeled and Unlabelled. Labeled data is a designation for pieces of data that have been tagged with one or more labels identifying certain properties or characteristics, or classifications or contained objects. Labels make that data specifically useful in supervised machine learning setups. On the other hand, Unlabelled data isn't tagged with these labels. So, we used both supervised and unsupervised ML algorithms. All hyperparameters were found using grid search over cross validation.

A. Logistic regression

It's a classification machine learning technique to classify data points separated by planes or hyperplanes. It minimizes the logistic loss (a smooth form of 0–1 loss) to find an optimal plane that best separates the two classes of data points. [13]

a) Retailer's price monitoring dataset: penalty:"l2", C:1

b) Annthyroid dataset: penalty:"l2", C:1

B. SVM - Support Vector Machine

SVMs [8] are one of the powerful machine learning algorithms for classification, regression and outlier detection purposes. An SVM classifier builds a model that assigns new data points to one of the given categories. Thus, it can be viewed as a non-probabilistic binary linear classifier.

a) Retailer's price monitoring dataset: 'C': 100, 'gamma': 1, 'kernel': 'rbf'

b) Annthyroid dataset: 'C': 1000, 'gamma': 0.1, 'kernel': 'rbf'

C. KNN - K-Nearest Neighbor

The k-nearest neighbors algorithm, also known as KNN [10] or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point with the assumption that similar points can be found near one another.

a) Retailer's price monitoring dataset: n_neighbors = 17

b) Annthyroid dataset: n_neighbors = 3

D. Multilayer Perceptron

A multilayer perceptron (MLP) [9] is a feed forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input nodes connected as a directed graph between the input and output layers. MLP uses backpropagation for training the network.

a) Retailer's price monitoring dataset: 'alpha': 0.01, 'hidden_layer_sizes': 9, 'max_iter': 600, 'random_state': 0, 'solver': 'lbfgs'

b) Annthyroid dataset: 'alpha': 0.001, 'hidden_layer_sizes': 5, 'max_iter': 300, 'random_state': 0, 'solver': 'lbfgs'

E. Decision Tree

Decision Tree [14] is one of the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

a) Retailer's price monitoring dataset: 'ccp_alpha': 0.001, 'criterion': 'entropy', 'max_depth': 9, 'max_features': 'auto'

b) Annthyroid dataset: 'ccp_alpha': 0.001, 'criterion': 'entropy', 'max_depth': 9, 'max_features': 'sqrt'

F. Random Forest

Random Forest [15] is a popular machine learning algorithm that belongs to the supervised learning technique. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

a) Retailer's price monitoring dataset: 'criterion': 'entropy', 'max_depth': 7, 'max_features': 'log2', 'n_estimators': 100

b) Annthyroid dataset: 'criterion': 'gini', 'max_depth': 8, 'max_features': 'auto', 'n_estimators': 500

G. XGBoost

The XGBoost (eXtreme Gradient Boosting) [7] is an efficient open-source implementation of the gradient boosted trees algorithm. It's a supervised learning algorithm that attempts to accurately predict a target variable by combining an ensemble of estimates from a set of simpler and weaker models.

a) Retailer's price monitoring dataset: 'learning_rate': 0.015, 'max_depth': 5, 'n_estimators': 200

b) Annthyroid dataset: 'learning_rate': 0.015, 'max_depth': 5, 'n_estimators': 200

H. One class SVM

One-Class Support Vector Machine (SVM) [16] is an unsupervised model that learns the boundary for the normal data points and identifies the data outside the border to be anomalies. Since the one-class SVM does not have target labels for the model training process unlike the regular supervised SVM,

we retrieve the outlier percentage as the prediction results for visualization.

I. Isolation Forest

Isolation Forests (IF) [17], like Random Forests, are built using decision trees. It is also an unsupervised model because there are no predefined labels. Isolation Forests is faster than one class SVM since randomly sub-sampled data is processed in a tree structure based on randomly selected features. The samples that travel deeper into the tree are less likely to be anomalies as they require more cuts to isolate them.

J. DBSCAN - Density-based spatial clustering of applications with noise

DBSCAN [18] is a spatial clustering technique that uses density to define anomalies in data sets. The fundamental concept behind the DBSCAN algorithm is to locate high-density regions that are separated from each other by low-density regions. It is a simple density algorithm that implements the notion of density by means of a center-based procedure.

K. Prophet

Prophet [11] is an open source univariate (one variable) time series forecasting algorithm designed by Facebook for ease of use without any expert knowledge in statistics or time series forecasting. It builds a model by finding a best smooth line which can be represented as a sum of the following components:

$$y(t) = g(t) + s(t) + h(t) + \epsilon \tag{1}$$

where g(t) : models trend, s(t) : models seasonality, s(t) : models the effects of holidays or large events, ϵ : is the error term.

L. DeepAR

DeepAR is a supervised auto-regressive learning algorithm for time series forecasting that uses recurrent neural networks (RNN) to produce both point and probabilistic forecasts. It's capable of handling multiple time-series in a single model creating a global model. [12]

III. Results

The results of supervised and Unsupervised algorithms are described in this section. I will display the decision boundaries of the supervised Retailer's price monitoring dataset as it has only 2 attributes and can be plotted in X-Y axis. The test results were conducted on the supervised data using 80-20 train test split.

A. Retailer's price monitoring dataset

The decision boundaries of the various ML techniques are displayed in Fig. 1. And the performance metrics are displayed in the table I. We can see from the table that XGBoost has the best result compared to other supervised classification models for this data.

TABLE I
RETAILER'S PRICE MONITORING DATASET RESULTS

Models	Test AUC	Test Acc.	Test F-1	Conf. Matrix
XGBoost	**0.9676**	**97.53**	**0.9736**	3004 21 57 84
SVM	0.9239	97.25	0.9678	3018 7 80 61
KNN	0.9315	97.5	0.9732	3004 21 58 83
MLP	0.9595	97.44	0.9719	3008 17 64 77
Decision Tree	0.9483	97.44	0.972	3007 18 63 78
Random Forest	0.9627	97.22	0.9684	3012 13 75 66
Logistic Reg.	0.9121	96.36	0.9568	3007 18 97 44

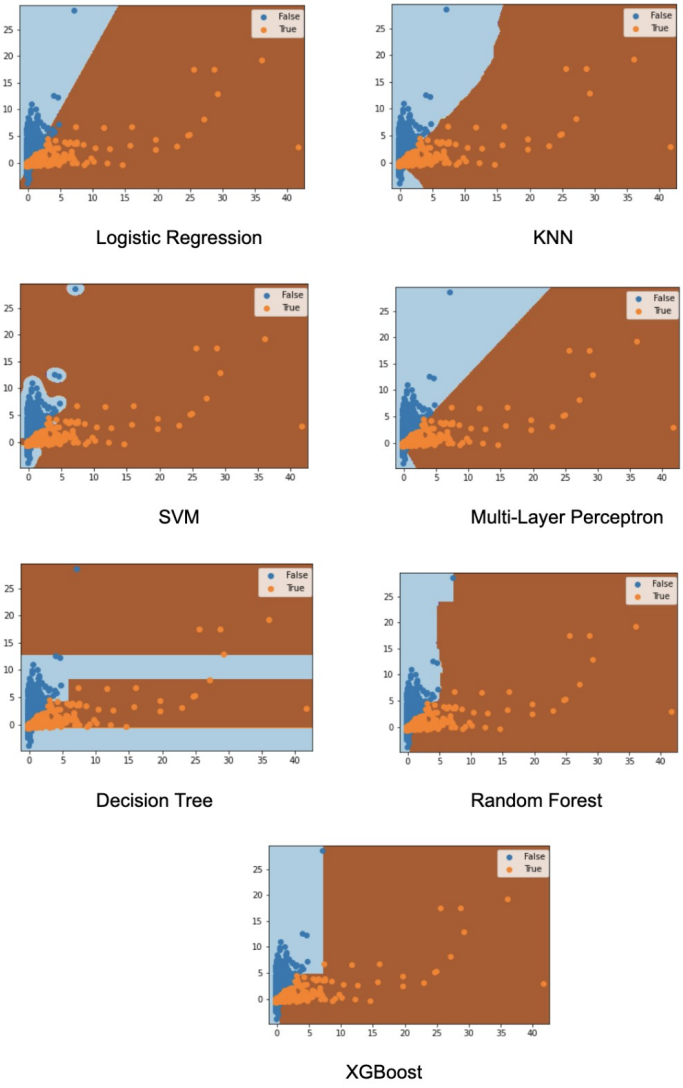

Fig. 1. Supervised Retailer's price monitoring dataset Decision Boundaries. Orange data points are Outliers.

B. Annthyroid dataset

The performance metrics are displayed in the table II. We can see from the table that XGBoost has the best result compared to other supervised classification models for this data.

TABLE II
ANNTHYROID DATASET RESULTS

Models	Test AUC	Test Acc.	Test F-1	Conf. Matrix
XGBoost	**0.9959**	**98.54**	**0.9859**	[1301 21; 0 118]
SVM	0.9762	96.52	0.9642	[1304 18; 32 86]
KNN	0.8391	95.06	0.9446	[1311 11; 60 58]
MLP	0.993	98.19	0.9826	[1298 24; 2 116]
Decision Tree	0.9853	97.98	0.9803	[1301 21; 8 110]
Random Forest	0.9963	98.54	0.9858	[1303 19; 2 116]
Logistic Reg.	0.9564	94.37	0.9323	[1317 5; 76 42]

C. Walmart's sales dataset

a) KNN: The model was instantiated by setting the value for K = 100 and the following distance matrix was created as shown in Fig. 2. The plot has x-axis with date values and the y-axis with the average distance of the point from the nearest k points. In order to determine the cutoff value, distances more than 50,000 points were taken and the following graph was plotted, the plot is visualized in Fig. 2.

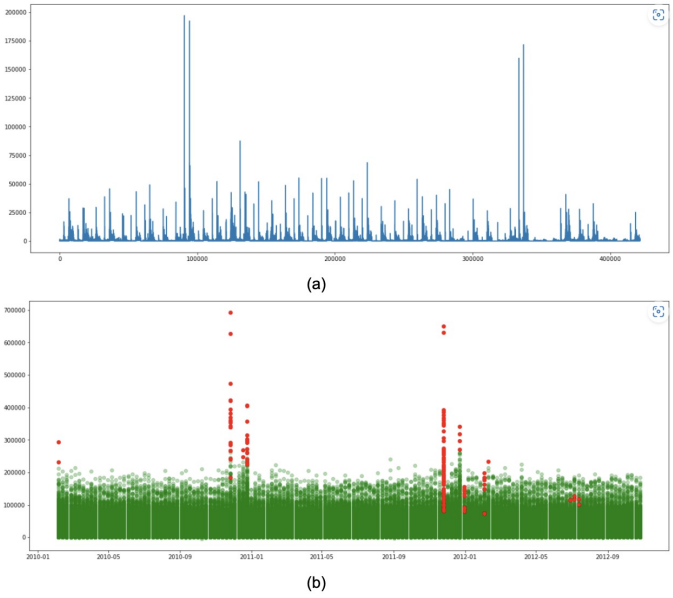

Fig. 2. KNN: (a) Distance graph of each data points, (b) Outlier graph showing outliers data with red color

b) One class SVM: We retrieve the results as the total percentage of the data that is outlier and visualize via a pie chart and bar graph as shown in Fig. 3.

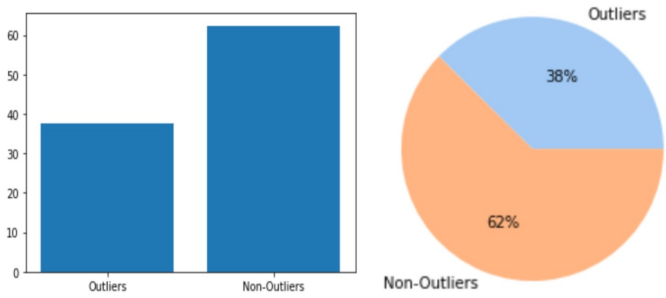

Fig. 3. One class SVM : Percentage of the Outliers and Non-Outliers

c) Isolation Forest: The algorithm begins with data training by generating Isolation Trees. Isolation Forests assume a small percentage of Outliers and hence we visualize the outlier percentage. We define the label anomaly to determine the outliers from the data and visualize using a pie chart as shown in Fig. 4.

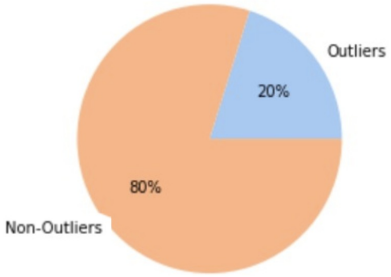

Fig. 4. Isolation Forest : Percentage of the Outliers and Non-Outliers

d) DBSCAN: The clusters after running the DBSCAN model with optimal parameters (eps = 2.0, minPts = 12) are visualized using the top 2 principal components in Fig. 5. We obtain a total of 17 clusters in which the points with Label 0 (Red) are the outliers and Label 1 - Label 16 are the non-outlier points.

e) Prophet: One of the forecasted time series (Store x Dept.) after training the model could be seen in Fig. 6. The black dots are the actual target values of the data. The solid blue curve represents the predicted values. The dotted red lines are the checkpoints. The blue background behind the curve represents the uncertainty level. The black dots outside the uncertainty level are the outliers for this Store X Dept. time series data. The figure has 5 outliers.

f) DeepAR: One of the forecasted time series(Store x Dept.) after training the model could be visualized in Fig. 7. Here, NLL is the negative log likelihood loss. The solid blue line represents the actual target values. The blue background behind the curve is the uncertainty level. The red line pointing to the target value is the outlier in the image.

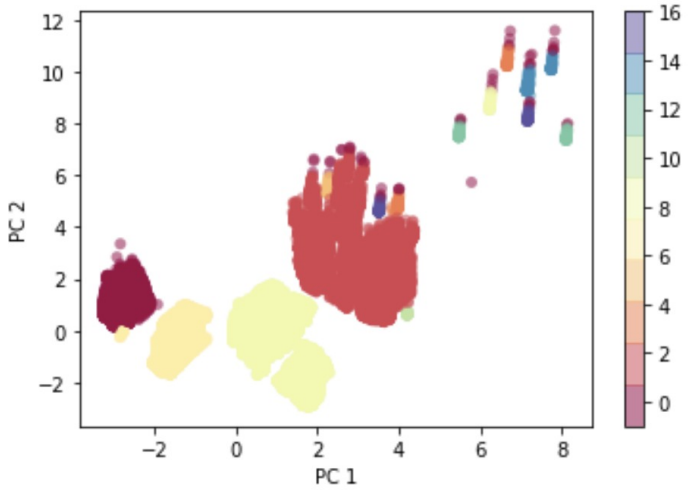

Fig. 5. DBSCAN clusters, Label 0 denote outliers, Label 1 - 16 are non-outliers.

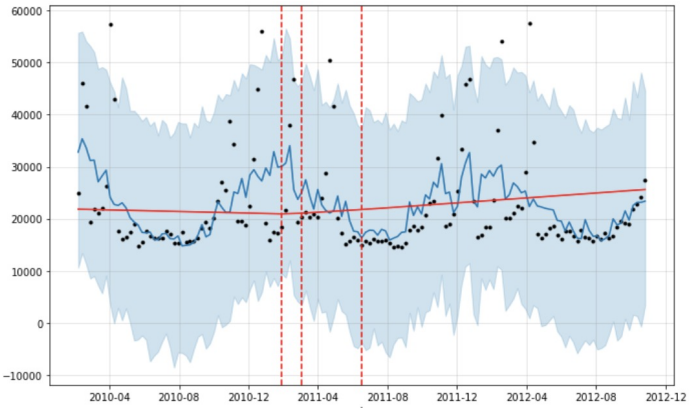

Fig. 6. Forecasted Store X Dept. result after training the Prophet model

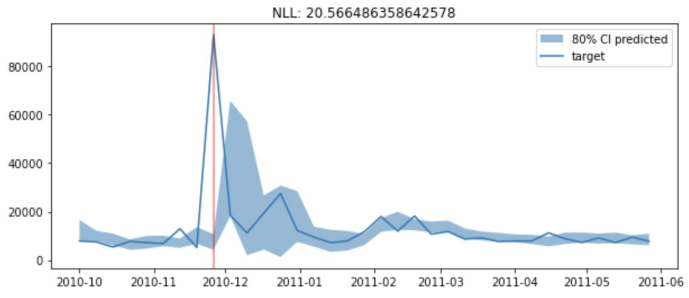

Fig. 7. Forecasted Store X Dept. result after training the DeepAR model

IV. Project Contribution

This is a Group project and my other team members include Riyank Mukhopadhyay, Shivani Madhunala, Shahana Ragupathy Sumathy, and Vagisha Gupta. My role was to research the problem statement, data preprocessing, modeling, and analyzing the results.

I worked independently on building supervised models for the anomaly detection namely Logistic regression, KNN, SVM, MLP, Decision Tree, Random Forest, and XGBoost classifier. In addition to the performance metrics, I also worked on plotting its decision boundaries.

In the unsupervised part, I worked end-to-end from data preprocessing to analyzing the results on Prophet and DeepAR. These are the time series models that required a different data preprocessing compared to other models described in this paper.

For the extra work in the project portfolio, I added an additional supervised data - The annthyroid dataset. In the modeling part, I added a few other supervised classification models - Logistic Regression, Decision Tree, and Random Forest. I also tuned the hyperparameters of all the supervised models using Grid search over cross-validation.

V. Knowledge Acquired

Implementing the supervised ML classifiers, I got the opportunity to work on the model parameters and hyperparameters very closely. In the data preprocessing part, I understood the importance of normalizing the data, data imputation, and stratified data sampling. In the modeling section, the hyperparameter tuning helped me realize how those minimal value changes affect the results. In the results, I learned the different performance metrics involved when the data classes are imbalanced.

Working on the time series models - Prophet and DeepAR, consumed most of my time because of the complexities involved in data preprocessing and modeling. DeepAR takes input in a particular format known as ListDataset. Also, while Prophet builds 1 model for each time series data, DeepAR builds 1 model for all the time series data. There isn't much work done in anomaly detection using DeepAR, so it took me some time studying and experimenting with it. While doing those experiments, I got the opportunity to rectify one of the bugs in the source code of GluonTS - DeepAR.

VI. Conclusion and Future Work

DeepAR trains one model jointly over all of the time series whereas Prophet trains one model for each time series. DeepAr took 39 seconds to train the model whereas Prophet took 77 minutes (120 times of DeepAR training time).

Most of the time, we don't have the outlier labels. So, anomaly detection using an unsupervised model is more convenient and popular. In the context of the unsupervised Walmart data used, It makes more sense to use the time series models as they have some seasonality and trend with respect to the past values. To save computation and memory, it is ideal to use the DeepAR. As it takes less time to train and saves disk

space. It is complex to handle the DeepAR hyperparameters, as it generalizes across all time series. So, In future, we will look to analyze the hyperparameters complexity.

REFERENCES

[1] A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," in IEEE Access, vol. 9, pp. 78658-78700, 2021, doi: 10.1109/ACCESS.2021.3083060

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 71–97, 2009, doi: 10.1145/1541880.1541882.

[3] M. Injadat, F. Salo, A. B. Nassif, A. Essex, and A. Shami, "Bayesian optimization with machine learning algorithms towards anomaly detection," in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647714. VOLUME 9, 2021 78691 A. B. Nassif et al.: Machine Learning for Anomaly Detection.

[4] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, Unsupervised Anomaly Detection With Generative Adversarial Networks to Guide Marker Discovery, vol. 10265, no. 2. Cham, Switzerland: Springer, 2017.

[5] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," Comput. J., vol. 54, no. 4, pp. 570–588, Apr. 2011, doi: 10.1093/comjnl/bxr026.

[6] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," Procedia Comput. Sci., vol. 60, no. 1, pp. 708–713, 2015, doi: 10.1016/j.procs.2015.08.220.

[7] Ikram, Sumaiya Thaseen, et al. "Anomaly detection using XGBoost ensemble of deep neural network models." (2021).

[8] Erfani, Sarah M., et al. "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning." Pattern Recognition 58 (2016): 121-134.

[9] Gardner, Matt W., and S. R. Dorling. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences." Atmospheric environment 32.14-15 (1998): 2627-2636.

[10] Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model-Based Approach in Classification. Alexandrov A, Benidis K, Bohlke-Schneider M, Flunkert V, Gasthaus J, Januschowski T, Maddix DC, Rangapuram SS, Salinas D, Schulz J, Stella L. GluonTS: Probabilistic and Neural Time Series Modeling in Python. J. Mach. Learn. Res.. 2020 Jan 1;21(116):1-6.

[11] Taylor SJ, Letham B. Forecasting at scale. The American Statistician. 2018 Jan 2;72(1):37-45.

[12] Alexandrov A, Benidis K, Bohlke-Schneider M, Flunkert V, Gasthaus J, Januschowski T, Maddix DC, Rangapuram SS, Salinas D, Schulz J, Stella L. GluonTS: Probabilistic and Neural Time Series Modeling in Python. J. Mach. Learn. Res.. 2020 Jan 1;21(116):1-6.

[13] Kleinbaum DG, Dietz K, Gail M, Klein M, Klein M. Logistic regression. New York: Springer-Verlag; 2002 Aug.

[14] Song YY, Ying LU. Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry. 2015 Apr 4;27(2):130.

[15] Breiman L. Random forests. Machine learning. 2001 Oct;45(1):5-32.

[16] Li KL, Huang HK, Tian SF, Xu W. Improving one-class SVM for anomaly detection. InProceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693) 2003 Nov 5 (Vol. 5, pp. 3077-3081). IEEE.

[17] Liu FT, Ting KM, Zhou ZH. Isolation forest. In2008 eighth ieee international conference on data mining 2008 Dec 15 (pp. 413-422). IEEE.

[18] Schubert E, Sander J, Ester M, Kriegel HP, Xu X. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS). 2017 Jul 31;42(3):1-21.

[19] https://www.kaggle.com/datasets/ujjwalchowdhury/walmartcleaned

[20] https://www.kaggle.com/competitions/anomaly-detection/data

[21] http://odds.cs.stonybrook.edu/annthyroid-dataset/