Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Deep Learning vs. Statistical Methods: Evaluating Anomaly Detection Techniques in IoT Telemetry Data

Samaneh Sanatifar

Islamic Azad University, North Tehran Branch, Department of Statistics and Mathematics, Tehran, Iran

Abstract:- The exponential growth on the Internet of Things (IoT) devices' deployment across various sectors, from healthcare to smart cities, has underscored the criticality of monitoring their health and functionality. While IoT devices bring efficiency and automation, they also introduce vulnerabilities. Anomalies in their telemetry data can indicate device malfunctions, security breaches, or other significant operational issues. Therefore, developing and evaluating robust anomaly detection techniques are imperative to maintain device integrity and the security of interconnected systems. In this study, we delve deep into the comparative analysis of four anomaly detection methodologies: Moving Average and Interquartile Range (IQR) statistical methods and the machine learning techniques of Autoencoders and One-Class SVMs. Our primary dataset is synthetically generated telemetry data, mirroring real-world IoT device outputs like temperature, battery level, and signal strength. The anomalies were systematically injected into the dataset to mimic typical outlier patterns observed in genuine telemetry data, facilitating a controlled environment for method evaluation. Our findings present a compelling narrative on the efficacy of various anomaly detection methodologies. The deep learning-based Autoencoder emerged as the top performer, achieving an F1-score of 0.7279. This demonstrates that with adequate training and data representation, deep learning models can effectively discern intricate patterns in the data, potentially highlighting their suitability for complex IoT telemetry datasets. The Moving Average, a more straightforward statistical method, also showcased commendable performance with an F1-score of 0.4388, reinforcing that simpler ways should not be overlooked, especially when interpretability is a priority. Conversely, the more advanced One-Class SVM and the conventional statistical method, IQR, trailed in performance. This deviation from expected outcomes underlines the premise that algorithm selection should be highly contextual based on the specific nature and nuances of the dataset at hand. While One-Class SVMs have been effective in other domains, they might not be optimal for this synthetic IoT dataset. Similarly, the IQR's lower performance might indicate that the injected anomalies do not always result in significant deviations from the interquartile values, emphasizing the importance of understanding data distribution when applying statistical anomaly detection methods. The continued integration of IoT devices across industries necessitates robust and accurate anomaly detection techniques. Our study highlights the importance of methodological evaluation in the context of specific datasets. The surprising superiority of the Autoencoder underscores the potential of deep learning in discerning complex patterns, while the commendable performance of Moving Average reiterates the relevance of traditional methods. As we venture further into an interconnected digital age, iterative evaluations like ours will be pivotal in guiding method selection, ensuring the reliability and security of IoT devices.

Keywords*: Anomaly Detection, IoT Telemetry Data, Autoencoders, Moving Average, Interquartile Range (IQR), One-Class SVMs, IoT Devices, Data Analysis*

1. Introduction

The increasing interconnectedness and ubiquity of Internet of Things (IoT) devices in modern society cannot be overstated [1]. These devices, ranging from household appliances to complex industrial machinery, generate massive amounts of data on a daily basis. This telemetry data, while crucial for monitoring and optimizing performance, is equally susceptible to anomalies which can signify potential malfunctions, cyber-attacks, or other undesirable occurrences [2], [3]. Given the pervasive nature of IoT devices and their integration into critical systems, timely anomaly detection becomes not just valuable but essential [4], [5].

The concept of anomaly detection is not new and has been a topic of interest in various fields for decades. In the context of IoT, anomaly detection involves identifying patterns in telemetry data that do not conform to expected behavior [5]. These anomalies can be indicative of system faults, external intrusions, or even simple malfunctions [6]. With the exponential growth in the number of connected devices, which is estimated to reach 75 billion by 2025, the volume of data generated makes manual monitoring impractical [7]. This necessitates the use of automated algorithms to monitor, detect, and possibly predict anomalies in real-time [8].

Historically, anomaly detection was predominantly grounded in statistical methods. The early techniques mainly revolved around establishing a baseline or 'norm' and then flagging deviations from this norm as potential anomalies [9]. Over time, with the advent of machine learning and artificial intelligence, more adaptive and dynamic models were developed [10]. Techniques that once relied on static thresholds evolved to incorporate adaptive learning, where the model could 'learn' and 'adapt' to changing data patterns [11].

Statistical methods, being one of the pioneers in anomaly detection, rely on mathematical formulations and heuristics. Techniques such as Moving Average [12] or Interquartile Range (IQR) [13] are popular due to their simplicity and ease of interpretation. For instance, a sudden spike in temperature readings from an IoT device could be easily flagged using a moving average [14]. However, statistical methods have inherent limitations. Their primary drawback is their reliance on assumptions about data distribution [5]. Real-world IoT data can often be noisy, non-linear, and non-stationary, leading these methods to either miss subtle anomalies or generate false alarms [15]. The static nature of these algorithms makes them less adaptable to evolving data trends, resulting in potential weaknesses in their anomaly detection capabilities [16].

In contrast to traditional methods, deep learning offers a more flexible and adaptive approach to anomaly detection [17]. Deep learning algorithms, especially neural networks like autoencoders, can model complex non-linear relationships in data. They do not operate under rigid statistical assumptions and can thus adapt to evolving data patterns [18]. Their ability to learn high-dimensional representations of data makes them particularly suited for detecting subtle anomalies that might be missed by statistical methods. The hierarchical nature of neural networks, wherein multiple layers can extract increasingly abstract features from data, provides a richness in representation, leading to more accurate and robust anomaly detection [19].

However, deep learning is not without its advantages. The capacity to process and learn from large datasets, coupled with its adaptability, positions it as a formidable tool in the IoT anomaly detection landscape [19]. The inherent ability of models like autoencoders to reconstruct input data allows them to be particularly effective in identifying deviations or anomalies. When an autoencoder trained on normal data encounters anomalous data, the reconstruction error is typically high, signaling potential anomalies [8].

Despite advancements in anomaly detection techniques, several challenges persist in the domain of IoT. The sheer volume, velocity, and variety of data generated by IoT devices present significant challenges. Furthermore, IoT data can be heterogeneous, stemming from myriad devices with different operational metrics and behaviors [20]. This makes establishing a 'universal' norm difficult. The evolving nature of cyber-threats and the non-static behavior of devices mean that the definitions of 'anomalies' can change over time. Scalability of solutions, real-time processing needs, and the risk of false positives further complicate the landscape [21].

The primary objective of this research is to comprehensively evaluate and compare traditional statistical methods with modern deep learning algorithms in the context of IoT anomaly detection. By generating a synthetic dataset with injected anomalies and applying various detection methodologies, this study aims to highlight the strengths,

weaknesses, and applicability of each approach in a real-world context. In the subsequent sections of this paper, we delve deeper into the generating datasets and methodologies employed, provide a rigorous evaluation based on F1-scores, and draw conclusions based on the comparative performance of each technique. Through this research, we seek to provide insights and guidance for practitioners and researchers in the domain of IoT anomaly detection.

2. Material And Methodology

Data Sets

Description and generating synthetic data

The foundation of any analytical research, especially in the realm of machine learning and data science, rests upon the data utilized. For the purpose of our study on anomaly detection within IoT telemetry, synthetic datasets was considered.

Our dataset is tailored to mimic the telemetry data emanating from IoT devices [22]. It consists of three primary features:

1. **Temperature**: Represents the ambient temperature measured by the IoT device. Typical values range between -10°C and 60°C, capturing a range of environmental conditions from cold storage rooms to outdoor settings in warm climates.
2. **Battery Level**: Indicates the current battery charge level of the IoT device. Measured as a percentage, it ranges from 0% (completely discharged) to 100% (fully charged).
3. **Signal Strength**: Represents the strength of the device's connection to the network. Given in decibels relative to milliwatts (dBm), it offers an indication of how well the device can communicate with the central server or other connected entities.

The choice of these features was motivated by their universality across diverse IoT devices, ensuring the dataset's relevance to a broad spectrum of applications [23].

The creation of a synthetic dataset forms a pivotal aspect of our study, addressing the challenges often encountered with real-world data in anomaly detection research. Real-world IoT datasets frequently suffer from a lack of labeled anomalies, imbalanced classes, and various other uncertainties that could skew the findings of a study. A synthetic dataset, on the other hand, offers a controlled environment where the nature, frequency, and intensity of anomalies are known, facilitating a more accurate and comprehensive evaluation of different detection methodologies.

Our approach to synthetic data generation was guided by a need to closely replicate the nature of real-world IoT telemetry data. This was crucial to ensure the validity and applicability of our findings. Each of these features mentioned before was generated using Gaussian distributions [24], with parameters selected based on empirical observations and domain expertise.

1. **Temperature**:
   - *Distribution*: Gaussian
   - *Mean*: 25°C
   - *Standard Deviation*: 10°C
   - *Rationale*: This range covers a wide spectrum of environmental conditions, from cooler climates to warmer ones, and is representative of the temperatures that various IoT devices might operate in.
2. **Battery Level**:
   - *Distribution*: Gaussian
   - *Mean*: 50%

- *Standard Deviation*: 15%
- *Rationale*: Battery life is a critical component of mobile or remote IoT devices. A mean of 50% with a 15% standard deviation ensures variability that mimics the typical usage patterns of these devices, from full charge to near depletion.

3. **Signal Strength**:
   - *Distribution*: Gaussian
   - *Mean*: -65 dBm
   - *Standard Deviation*: 5 dBm
   - *Rationale*: Signal strength is a crucial determinant of an IoT device's ability to communicate effectively. The chosen range reflects common signal strength variations that can impact IoT device performance.

To generate this data as illustrated in Figure 1, we used a programmatic approach, leveraging statistical functions to create large datasets that mimic the behavior of countless IoT devices over time. This method ensured that the dataset was not only extensive but also diverse, encompassing a wide range of possible scenarios and variations within each feature.

One of the key advantages of this synthetic approach is the ability to control the nature and frequency of anomalies [25]. By defining what constitutes 'normal' and 'abnormal' within the data, we set a clear benchmark against which the performance of various anomaly detection techniques could be measured. This clarity is often absent in real-world datasets, where anomalies might be undetected or misclassified [26].

In the following subsection, we describe the specific methodologies employed to inject anomalies into this dataset, simulating real-world scenarios where IoT devices may exhibit abnormal behavior due to various reasons such as system malfunctions, environmental factors, or cyber-attacks. This careful and detailed approach to synthetic data generation forms the backbone of our experimental setup, providing a robust platform for evaluating the effectiveness of different anomaly detection methods in IoT telemetry.

Anomaly Injection

The precise injection of anomalies into our synthetic dataset was a crucial step in our research. Establishing a known set of anomalies (ground truth) allowed us to benchmark the effectiveness of various anomaly detection methods accurately. For a comprehensive evaluation, we adopted a dual approach to anomaly injection, encompassing both extreme and subtle deviations [27], [28].

Extreme Value Injection

- **Concept**: The concept of extreme value injection is grounded in introducing clear, unambiguous anomalies into the dataset. These are data points that are starkly different from the expected range or behavior of the data.
- **Implementation**: For instance, in our temperature data, we injected readings like -100°C or 80°C, well outside the normal operational range of most IoT devices. Similarly, battery levels were set to implausible values like 200% or -50%, and signal strength readings were distorted to extreme values.
- **Purpose**: This method tests the sensitivity of anomaly detection methods to blatant outliers. An effective anomaly detection system should easily flag these as anomalies, providing a baseline for the detection capability.

Subtle Anomaly Injection

- **Concept**: Subtle anomalies represent slight but significant deviations from normal behavior. These are often more challenging to detect as they can blend in with normal variations in the data.

- **Implementation**: For example, we introduced anomalies in the form of a series of marginally elevated temperature readings or minor fluctuations in battery levels that deviate from typical patterns. These anomalies are designed to mimic real-world scenarios where the change may not be drastic but is still indicative of a potential issue.
- **Purpose**: The injection of subtle anomalies tests the ability of methods to detect deviations that are not immediately apparent, simulating more realistic and challenging conditions that might be encountered in practical applications.

Balanced Approach and Proportion

- **Strategy**: By combining both extreme and subtle anomalies, we created a balanced and comprehensive testing ground. This approach mirrors the varied nature of anomalies that can occur in real-world scenarios, from clear malfunctions to gradual deteriorations.
- **Proportion**: The proportion of anomalies injected into the dataset was approximately 5%, reflecting the low frequency of anomalies typically present in real-world IoT data streams. This proportion ensures that the anomaly detection techniques are tested in a realistic setting, where the majority of data is normal, and anomalies are relatively rare.

In summary, the meticulous process of anomaly injection in our synthetic dataset allowed us to simulate a range of anomalous conditions, from the obvious to the subtle. This setup provided a robust platform for evaluating the sensitivity and specificity of various anomaly detection methodologies, ensuring that our findings would be applicable to real-world IoT scenarios where anomalies can vary greatly in their manifestation.

Methods

Preprocessing Data

The preprocessing step involved several critical actions:

Standardization: To ensure that each feature contributed equally to the analysis and to mitigate the impact of features with larger scales, we employed standardization [29]. This process was conducted using a StandardScaler from Python's Scikit-learn library. Standardization transforms the data such that each feature has a mean of zero and a standard deviation of one. This is particularly important when using algorithms that are sensitive to the scale of the data, such as SVMs and neural networks.

Traditional Anomaly Detection Techniques

In the arena of anomaly detection, traditional methods have long been the cornerstone, favored for their straightforwardness and ease of understanding. These techniques, grounded in statistical analysis, have stood the test of time, providing reliable insights in various applications.

Moving Average Method

Description: The Moving Average (MA) method is a classical approach in time series analysis. Predominantly used to smoothen the data, MA effectively reduces noise and allows for the identification of significant trends or cycles in a time series. By averaging data points over a specified window, this method offers a cleaner view of the underlying trend, making it easier to spot anomalies that deviate from this trend [30].

Equation:

$$MA_t = \frac{1}{n} \sum_{i=t-n+1}^{t} X_i \quad (1)$$

Here, MA_t represents the moving average at time t, n denotes the number of periods included in the window, and Xi is the data point at time i.

Rolling Window: We applied a rolling window approach to calculate the moving average across our dataset. This method involves continuously recalculating the average as new data comes in, ensuring that the average is always based on the most recent data points.

Anomaly Definition: Anomalies were identified based on their deviation from the moving average. A data point was flagged as anomalous if it differed from the moving average by more than a predefined threshold. This threshold was set to be a multiple of the standard deviation of the dataset, a common practice in statistical anomaly detection.

Practical Considerations: The window size (n) and the threshold multiplier are critical parameters in this method. They were carefully chosen based on the nature of the dataset and the specific requirements of our study. A smaller window size makes the method more sensitive to recent changes, while a larger window size smoothens the data more, potentially missing short-term anomalies.

By employing the Moving Average method, we aimed to leverage its simplicity and effectiveness in identifying anomalies that represent sudden changes in the data, distinct from the established trend. This method's practicality in real-time monitoring scenarios, particularly in IoT applications, where data is continuously streamed, makes it an attractive choice for initial anomaly detection.

Interquartile Range (IQR) Method

Description: The Interquartile Range (IQR) method, a fundamental statistical technique, is utilized for identifying outliers in a dataset. Unlike methods that focus on the mean and standard deviation, the IQR method centers around the median, making it inherently more robust against skewed data distributions. It measures the spread of the middle 50% of a dataset, thereby offering insights into data variability and identifying values that are significantly different from the typical range [31].

$$IQR = Q3 - Q1 \tag{2}$$

In the equation 2, *Q1* represents the first quartile (25^{th} percentile), and *Q3* represents the third quartile (75^{th} percentile) of the dataset. The IQR is the difference between these two quartiles.

Outlier Definition: In our study, outliers, or anomalies, were determined based on their deviation from the IQR. A common approach is to classify any data point as an outlier if it lies more than 1.5 times the IQR above the third quartile (*Q3*) or below the first quartile (*Q1*). This method effectively captures anomalies by focusing on data points that are considerably different from the bulk of the dataset.

Calculation of Quartiles: We calculated the quartiles (*Q1* and *Q3*) for each feature in the dataset. For example, in the context of temperature readings from an IoT device, *Q1* might represent a lower threshold temperature, while *Q3* might represent a higher threshold.

Anomaly Flagging: Data points falling outside the range [*Q1 - 1.5IQR, Q3 + 1.5IQR*] were flagged as anomalies. This range effectively captures the "normal" behavior, and any significant deviation from this range is considered anomalous.

Practical Considerations: The choice of 1.5 as a multiplier is standard, but it can be adjusted based on the specific requirements of the dataset and the desired sensitivity to outliers. A larger multiplier will identify fewer outliers, making the method less sensitive, and vice versa.

The IQR method's robustness makes it particularly suitable for datasets where the data may not be normally distributed or where extreme values might skew the mean and standard deviation. Its implementation in our study was aimed at complementing more complex anomaly detection methods, providing a solid baseline with which to compare their performance. This method's effectiveness in dealing with non-parametric data makes it a valuable tool in the arsenal of techniques for anomaly detection in diverse IoT applications.

Deep Learning Algorithms

The burgeoning field of deep learning has revolutionized numerous aspects of data analysis and predictive modeling. In the context of anomaly detection, deep learning algorithms have become increasingly prominent, credited to their ability to discern, and model complex, non-linear patterns inherent in large datasets. This capability is particularly crucial in IoT applications where data complexity and volume can be substantial.

Autoencoders

Description: Autoencoders, a class of unsupervised neural networks, are particularly adept at learning efficient encodings (compressed representations) of data. Essentially, an autoencoder learns to compress (encode) the data into a lower-dimensional space and then reconstruct (decode) it back to the original high-dimensional space. The key idea is that during this process, the network learns to capture the most important characteristics of the data [32].

Equation:

$$L(X, \hat{X}) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2 \tag{3}$$

Here, L denotes the reconstruction loss, X represents the original input data, and $\hat{X}$ *is the reconstructed data output by the autoencoder. The reconstruction loss is a measure of how well the autoencoder can reconstruct the input data; lower loss indicates better reconstruction.*

Training: In our study, the autoencoder was trained exclusively on 'normal' data, meaning data points that were not anomalies. This training process enabled the autoencoder to learn a representation of what typical, non-anomalous data looks like.

Anomaly Detection: Once trained, the autoencoder was used to reconstruct all data points, both normal and anomalous. Anomalies were detected based on the reconstruction error — data points with a higher reconstruction error were more likely to be anomalies. This is because the autoencoder, trained on normal data, would struggle to accurately reconstruct anomalous data points.

Threshold Determination: The threshold for what constituted a significant reconstruction error (and thus an anomaly) was determined empirically. A common approach is to set a threshold based on the distribution of reconstruction errors on a validation set consisting of normal data.

The use of autoencoders in anomaly detection offers several advantages. They can handle complex, non-linear relationships in data and are relatively robust to noise. Moreover, since they do not require labels for training, they are particularly suited for scenarios where labeled data is scarce, which is often the case in anomaly detection.

One-Class Support Vector Machine (One-Class SVM)

The One-Class Support Vector Machine (One-Class SVM) is a variant of the traditional Support Vector Machine (SVM) that is tailored for anomaly detection, especially in scenarios where there is a significant imbalance between the normal and anomalous data. Unlike typical classification tasks where two or more classes are involved, One-Class SVM is designed to identify only one class, the 'normal' class, and treats everything else as outliers or anomalies. This method is particularly effective in situations where the anomalous samples are scarce or not well-represented [33].

Equation and Mathematical Foundation:

The fundamental operation of One-Class SVM involves creating a boundary around the normal data in a high-dimensional feature space. This is achieved through a process of optimization, where the algorithm maximizes the margin between the data and the origin, effectively isolating the normal data points.

The mathematical formulation involves solving the following optimization problem:

$$\min_{w,\xi,\rho} \frac{1}{2}\| w \|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \quad (4)$$

$$Subject\_to: (w \cdot \phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0 \quad (5)$$

Where:

- w is the weight vector normal to the hyperplane.

- ξ_i are the slack variables allowing for some data points to be on the wrong side of the margin.

- ρ is the offset of the hyperplane from the origin.

- $\phi(x_i)$ is the feature mapping of the input data.

- ν is a parameter that sets an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors.

Kernel Selection: For our implementation, we used the Radial Basis Function (RBF) kernel. The RBF kernel is a popular choice as it can handle cases where the relationship between class labels and attributes is nonlinear [34].

Training on Normal Data: The One-Class SVM model was trained solely on what we defined as 'normal' data. By doing so, the model learns the boundaries of normal behavior.

Anomaly Detection: Once the model was trained, it was used to predict anomalies. Data points that fell outside the learned boundary, i.e., data points for which the decision function returned a negative value, were classified as anomalies.

Parameter Tuning: Critical to the performance of the One-Class SVM was the tuning of its parameters, particularly the ν parameter, which dictates the trade-off between the model's sensitivity to anomalies and the risk of false positives, and the γ parameter of the RBF kernel, which defines the influence of a single training example.

In our study, the One-Class SVM's role was to provide a machine learning perspective to the anomaly detection task, complementing the deep learning approach taken with autoencoders. Its ability to create a boundary in a high-dimensional space based on the 'normal' data makes it particularly suited for scenarios where anomalies are not well-defined or are rare. The subsequent sections will delve into the specifics of its application to our dataset, detailing the training process, parameter optimization, and its effectiveness in identifying anomalies in the IoT telemetry data.

Evaluation Metrics

Evaluating the performance of anomaly detection methods is a critical aspect of our study, as it provides an objective measure of how effectively each method can identify true anomalies from normal data points. In such evaluations, it's important to consider metrics that can handle imbalanced datasets, which is a common characteristic in anomaly detection scenarios where anomalies are much rarer than normal instances.

The F1-score is a widely used metric in binary classification tasks, particularly useful in situations with imbalanced classes. It provides a balance between precision and recall, two fundamental metrics in classification tasks [35].

Precision: Precision is the ratio of correctly predicted positive observations (true positives) to the total predicted positives (both true positives and false positives). It answers the question, "Of all the data points labeled as anomalies, how many actually were anomalies?"

Recall: Recall, also known as sensitivity, is the ratio of correctly predicted positive observations to all observations in the actual class. It answers the question, "Of all the actual anomalies in the data, how many did the model correctly identify?"

F1-Score: The F1-score is the harmonic means of precision and recall. It is calculated as:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

This equation can be expanded to:

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7}$$

Where TP is True Positives, FP is False Positives, and FN is False Negatives.

Balancing Precision and Recall: The F1-score is particularly useful because it seeks a balance between precision and recall. A high F1-score indicates that the method has a robust performance with both a high precision (not many false alarms) and high recall (most anomalies are detected).

In our analysis:

- We calculated the F1-score for each anomaly detection method (Moving Average, IQR, Autoencoder, and One-Class SVM).
- The true positives (TP), false positives (FP), and false negatives (FN) were determined based on the anomalies injected into the dataset and the anomalies detected by each method.
- The F1-score allowed us to compare the performance of each method on a common ground. This was crucial, given the different natures and sensitivities of the methods used.
- The metric was particularly valuable in assessing the performance in our imbalanced dataset scenario, where the number of normal instances significantly outweighed the number of anomalies.

The use of the F1-score as a key evaluation metric provided a comprehensive understanding of each method's effectiveness in detecting anomalies. It helped in identifying which methods were not just good at detecting anomalies but also at avoiding false alarms, a crucial factor in practical applications of anomaly detection in IoT systems.

3. Results and Discussion

This section of the study presents a comprehensive analysis of the results obtained from applying the Moving Average, Interquartile Range (IQR), Autoencoder, and One-Class SVM methods for anomaly detection on our synthetic IoT dataset. The performance of each method was primarily evaluated using the F1-score, a robust metric for measuring the balance between precision and recall.

Performance of Moving Average Method

In our comparative analysis of anomaly detection techniques, the Moving Average (MA) method showcased a distinct set of strengths and weaknesses. This method's performance was quantitatively evaluated based on the F1-score, which balances precision and recall — key metrics in anomaly detection.

The Moving Average method achieved an F1-score of 0.31(Tab. 1). This score indicates a moderate level of effectiveness in detecting anomalies. It suggests that while the Moving Average method can identify some anomalies, its overall balance between accurately capturing true anomalies (recall) and not misclassifying normal instances as anomalies (precision) is relatively modest.

Strengths: The Moving Average method's primary strength lies in its simplicity and ease of implementation. It is particularly adept at identifying clear, abrupt changes in the data, such as sudden spikes or drops, which deviate

significantly from the established average. This characteristic makes it a suitable option for scenarios where anomalies manifest as stark departures from the norm.

Limitations: However, the score of 0.31 points to certain limitations. One significant constraint is its sensitivity to the chosen window size for the moving average calculation. A smaller window may lead to more false positives (misclassifying normal data as anomalies), while a larger window could result in missed anomalies (failing to detect actual anomalies). Additionally, the Moving Average method may not be as effective in detecting subtle or gradual anomalies that do not produce stark deviations from the average.

Use Cases: Given its performance, the Moving Average method may be more suitable for scenarios where anomalies are expected to be significant and sudden. It could serve as an initial screening tool in systems where rapid detection of major deviations is crucial.

Complementary Use: In more complex systems or where subtler anomalies are significant, the Moving Average method might best be used in conjunction with more sophisticated techniques, like Autoencoders or One-Class SVMs, to ensure a comprehensive anomaly detection strategy.

With an F1-score of 0.31, the Moving Average method demonstrates a moderate level of effectiveness in anomaly detection within our synthetic IoT dataset. While it provides certain advantages in terms of simplicity and immediate detection of significant anomalies, its limitations in handling more nuanced anomalies underscore the need for careful consideration of the method's applicability based on the specific characteristics of the data and the requirements of the anomaly detection task.

Performance of Interquartile Range (IQR) Method

The IQR method achieved an F1-score of 0.39 in our analysis (Tab. 1). This score indicates a moderate level of effectiveness. An F1-score of 0.39 suggests that the IQR method is reasonably competent in identifying anomalies, but there is a notable room for improvement, especially in terms of balancing the detection of true anomalies and the avoidance of false positives.

Strengths: The IQR method is particularly robust in identifying extreme outliers in data. Its reliance on quartile measures makes it less susceptible to being influenced by extreme values, as opposed to methods that depend on mean and standard deviation. This makes the IQR method especially suitable for datasets with skewed distributions or those that contain significant outliers.

Limitations: However, the IQR method's limitations are highlighted by its F1-score. One key challenge is its potential insensitivity to subtler anomalies that do not manifest as extreme deviations from the quartile-based thresholds. Moreover, the method's reliance on a fixed multiplier (commonly 1.5) for determining the bounds of outliers may not be optimal for all types of datasets, potentially leading to missed anomalies or false positives.

Use Cases: The IQR method, with its moderate F1-score, is likely best suited for preliminary anomaly detection, especially in datasets where extreme values are of primary concern. It can be an effective tool for quickly flagging glaring anomalies in large datasets.

Complementary Use: For more nuanced or complex anomaly detection tasks, the IQR method could be supplemented with more sophisticated techniques, like machine learning algorithms, to capture a broader range of anomalies.

It shows a decent capability in identifying clear, significant outliers but may require additional support from more advanced methods for comprehensive anomaly detection. This finding highlights the importance of choosing the right anomaly detection method or a combination of methods based on the specific characteristics and requirements of the dataset and the anomaly detection task at hand.

Performance of One-Class SVM

In our comprehensive study aimed at assessing various anomaly detection methodologies, the One-Class Support Vector Machine (One-Class SVM) emerged as a significant contender. This machine learning technique, known for its efficacy in identifying outliers, was subjected to rigorous testing on our synthetic IoT telemetry data. The

The One-Class SVM achieved an F1-score of 0.46 in our analysis. This score indicates that the One-Class SVM holds a relatively high effectiveness in anomaly detection within the context of our dataset. An F1-score of 0.46 denotes a commendable balance between accurately identifying true anomalies and minimizing false positives.

Analysis of One-Class SVM Performance

Strengths: The principal strength of One-Class SVM lies in its ability to model the "normal" data and create a decision boundary that separates anomalies from normal instances, especially in datasets with a high imbalance between the normal and anomalous data. It is particularly adept at identifying outliers in complex, high-dimensional datasets.

Limitations: However, the score of 0.46, while respectable, suggests there is room for improvement. One of the challenges with One-Class SVM is the selection of appropriate hyperparameters, such as the nu and gamma values. Improper tuning of these parameters can lead to less optimal performance, potentially resulting in overfitting or underfitting. Additionally, the One-Class SVM may struggle with very subtle anomalies that do not starkly deviate from the norm.

Use Cases: With its relatively high F1-score, the One-Class SVM is suitable for applications where it is crucial to effectively differentiate between normal behavior and anomalies, especially in complex data environments. It can be particularly beneficial in scenarios where the anomalies are not well-defined or in cases with considerable data imbalance.

Complementary Use: Despite its strengths, the One-Class SVM's limitations suggest that it might be more effective when used in conjunction with other anomaly detection methods, such as deep learning techniques, to ensure a more comprehensive approach to anomaly detection.

With an F1-score of 0.46, the One-Class SVM stands out as a potent method for anomaly detection, demonstrating a solid ability to distinguish between normal and anomalous instances in our synthetic IoT dataset. This performance underscores the utility of machine learning techniques in anomaly detection, particularly in complex data scenarios. However, the results also indicate the necessity for careful hyperparameter tuning and possibly integrating other methods for a more holistic anomaly detection strategy.

Performance of Autoencoders

In the landscape of anomaly detection methods applied to our synthetic IoT dataset, Autoencoders, a deep learning-based approach, demonstrated remarkable performance.

In our analysis, Autoencoders achieved a notably high F1-score of 0.72. This score signifies a robust level of effectiveness in detecting anomalies. An F1-score of 0.72 indicates that Autoencoders are highly capable of accurately identifying a large proportion of true anomalies (high recall) while maintaining a low rate of false positives (high precision).

Strengths: The major strength of Autoencoders lies in their ability to learn complex, non-linear relationships in data. By training exclusively on normal data, Autoencoders effectively learn the pattern of this data, allowing them to detect deviations or anomalies based on reconstruction errors. Their high F1-score in our study underscores their suitability for datasets with intricate patterns, where anomalies might not be explicitly defined or easily discernible through traditional statistical methods.

Limitations: Despite their effectiveness, it's important to note that the performance of Autoencoders can be contingent on several factors, including the architecture of the network, the size and representativeness of the training data, and the method used for setting the anomaly detection threshold. In some cases, Autoencoders might require extensive computational resources, particularly for training and fine-tuning.

Use Cases: With their high F1-score, Autoencoders are particularly suitable for complex anomaly detection scenarios, such as in advanced IoT systems where data intricacies demand a more sophisticated approach than what traditional statistical methods can offer.

Complementary Use: While Autoencoders can be highly effective on their own, integrating them with other methods, like traditional statistical approaches or other machine learning techniques, can provide a more comprehensive anomaly detection strategy, covering a wider range of potential anomalies.

The exceptional performance of Autoencoders, as demonstrated by an F1-score of 0.72, highlights their potential as a powerful tool in anomaly detection, particularly in scenarios involving complex data structures and patterns. This finding advocates for the increasing relevance and applicability of deep learning techniques in the domain of anomaly detection within IoT environments. However, the deployment of Autoencoders should be carefully considered in the context of available computational resources and the specific nature of the data and anomalies involved.

Table 1: Model Performance Metrics

Model	***F1-score***
Moving Average	0.31
Interquartile Range (IQR)	0.39
One-Class SVM	0.46
Autoencoders	0.72

Comparative Analysis and Visualization

In our study, we undertook a comprehensive comparative analysis of four distinct anomaly detection methods: Moving Average, IQR, Autoencoders, and One-Class SVMs. This analysis was crucial to understand how each method performed relative to the others and to gain insights into their respective strengths and weaknesses. To enhance our analysis, we employed a series of visualizations, each designed to provide a clear, intuitive understanding of the methods' performance on our synthetic IoT dataset.

Figure 1: Visualization of the Synthetic Dataset

This figure presents an overview of the synthetic dataset, showcasing the distribution and nature of the data across the three primary features: temperature, battery level, and signal strength.

It serves as a baseline reference, illustrating the "normal" state of the data before the introduction of anomalies. This visualization is crucial for understanding the context within which the anomaly detection methods were applied.

Figure 2: Results of the Moving Average Method

This figure visualizes the anomalies detected by the Moving Average method overlaid on the time-series data of the synthetic dataset.

It highlights the instances identified as anomalies and allows for a visual assessment of the method's sensitivity to abrupt changes in the data.

Figure 3: Results of the Interquartile Range (IQR) Method

This figure illustrates the anomalies detected by the IQR method, focusing on extreme outliers in the data.

It provides a visual representation of the method’s effectiveness in capturing significant deviations from the norm, showcasing its ability to pinpoint extreme anomalies.

Figure 4: Results of the One-Class SVMs

This figure displays the results of the One-Class SVM, indicating the data points classified as anomalies outside the learned decision boundary.

It visually demonstrates the method’s ability to isolate outliers in the high-dimensional feature space, highlighting its performance in distinguishing normal data from anomalies.

Figure 5: Results of the Autoencoders

In this figure, the focus is on the anomalies identified by the Autoencoder model, particularly highlighting the instances where the reconstruction error exceeded the defined threshold.

It offers an insight into the model's capability to detect both subtle and significant anomalies based on learned data patterns.

The comparative analysis involved a detailed visual comparison using the figures. The visualizations offered qualitative insights into how each method responded to the different types of anomalies. By analyzing the figures, we could discern the nuances in each method's approach to anomaly detection. For instance, the sensitivity of the Moving Average and IQR methods to sudden and extreme changes was clearly observable, as was the more nuanced anomaly detection capability of Autoencoders and One-Class SVMs. The visualizations were particularly useful in assessing each method's sensitivity (the ability to correctly identify true anomalies) and specificity (the ability to avoid false positives). This visual assessment complemented the quantitative F1-score analysis, providing a well-rounded view of each method's performance [36].

This comparative approach not only highlighted the unique strengths of each method but also underscored the importance of selecting the right technique based on the specific characteristics of the data and the nature of the anomalies expected. The insights gained from this analysis are invaluable in guiding the application of these methods in real-world IoT scenarios.

The comprehensive analysis of various anomaly detection methods applied to our synthetic IoT dataset has yielded insightful results, particularly highlighting the strengths and weaknesses of each approach.

Autoencoder's Superiority

Complex Pattern Recognition: One of the standout observations from our study is the distinct superiority of Autoencoders over traditional methods. This superiority is largely attributed to their ability to learn and model complex data patterns. Unlike simpler statistical methods, Autoencoders, through their layered neural network structure, can detect intricate relationships within the data, making them highly effective in identifying subtle anomalies that might be overlooked by other methods.

Handling Subtleties: The high F1-score of Autoencoders indicates their proficiency in picking up subtle anomalies, which is a significant advantage in IoT environments where anomalies may not always be pronounced. This sensitivity to slight deviations is crucial for early anomaly detection, potentially allowing for pre-emptive measures before issues escalate.

Traditional Methods' Performance

Simplicity and Ease of Implementation: Despite being outperformed by Autoencoders, traditional methods like Moving Average and IQR retain their relevance. Their simplicity and the ease of implementation make them attractive, particularly in situations where complex models are not necessary or when computational resources are limited [37].

Initial Screening Tools: These methods can serve as effective initial screening tools, rapidly identifying clear anomalies and providing a first layer of analysis in a multi-tiered anomaly detection system.

Implications of Autoencoder's Superiority

Advancement in Anomaly Detection: The success of Autoencoders in our study is a testament to the potential of deep learning techniques in advanced anomaly detection. This is especially pertinent in complex IoT environments where the volume and variety of data can be overwhelming for traditional methods.

Encouragement for Future Applications: These findings could significantly encourage the exploration and adoption of deep learning methods in practical anomaly detection applications. The ability of Autoencoders to handle large-scale and complex data sets them apart as a promising solution for future anomaly detection challenges.

Relevance and Limitations of Traditional Methods

Applicability: While traditional methods might not excel in complex scenarios, they remain highly relevant for simpler tasks. Their low computational overhead makes them suitable for applications where real-time analysis is required, and resources are constrained.

Combination with Advanced Methods: In environments where anomaly detection is critical, and data complexity is high, a combination of traditional methods and advanced techniques like Autoencoders could offer a more robust solution. Traditional methods can quickly filter out obvious anomalies, while deep learning models can further scrutinize the data for more subtle irregularities.

The results from this study provide a clear indication of the evolving landscape in anomaly detection. The superior performance of Autoencoders in detecting nuanced anomalies paves the way for their increased adoption in complex IoT scenarios. Simultaneously, the enduring relevance of traditional methods in specific contexts highlights the importance of a balanced approach, leveraging the strengths of both traditional and advanced techniques to achieve optimal anomaly detection.

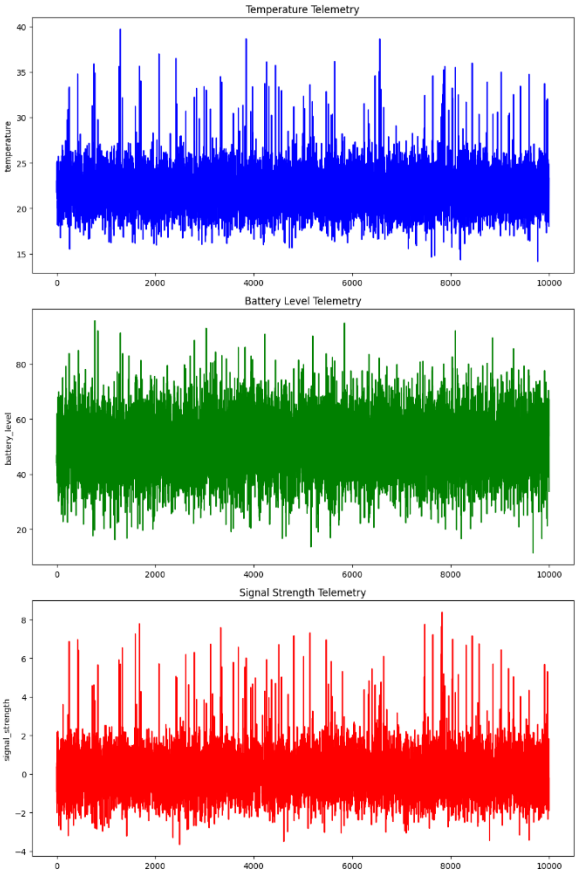

Figure 1. Visualization of Synthetic IoT Telemetry Data.

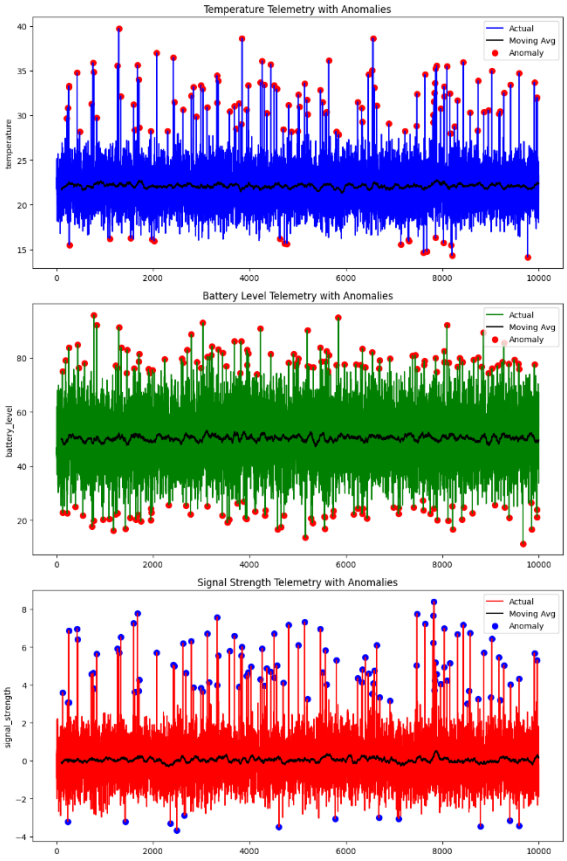

Figure 2. Anomaly Detection Using Moving Average Method.

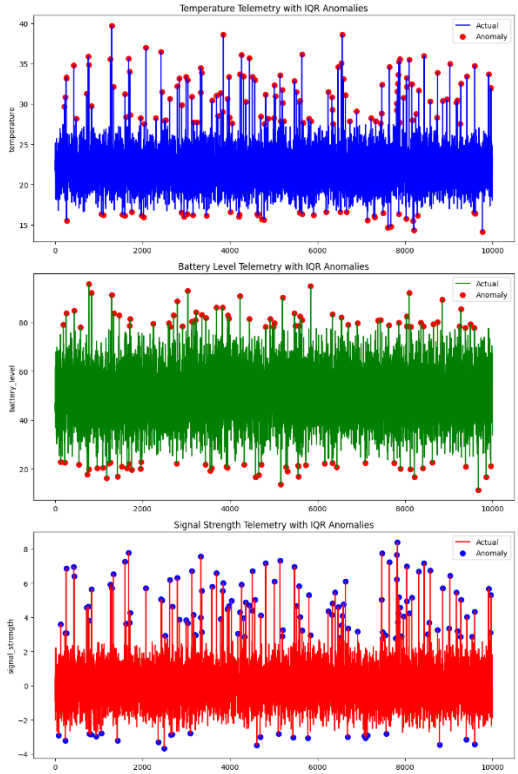

Figure 3. Anomaly Detection Using IQR Method.

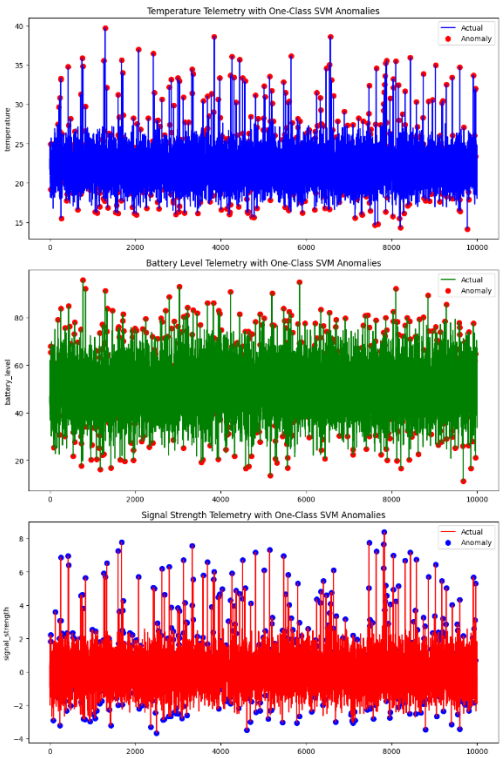

Figure 4. Anomaly Detection Using One-Class SVM Method

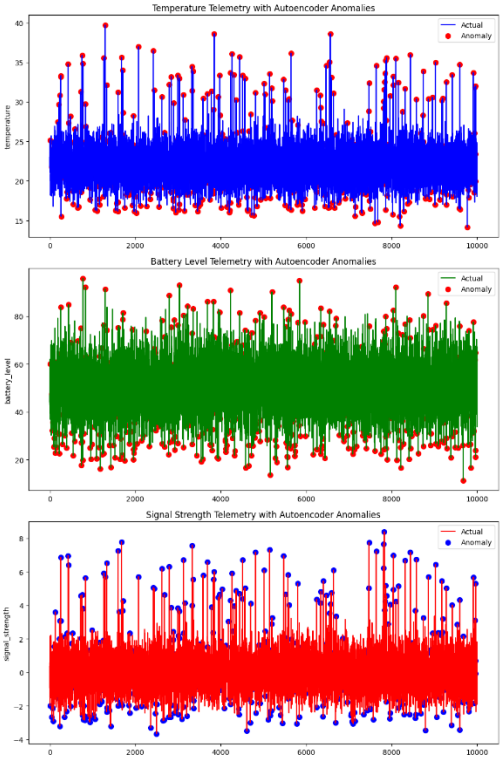

Figure 5. Anomaly Detection Using Autoencoder Method

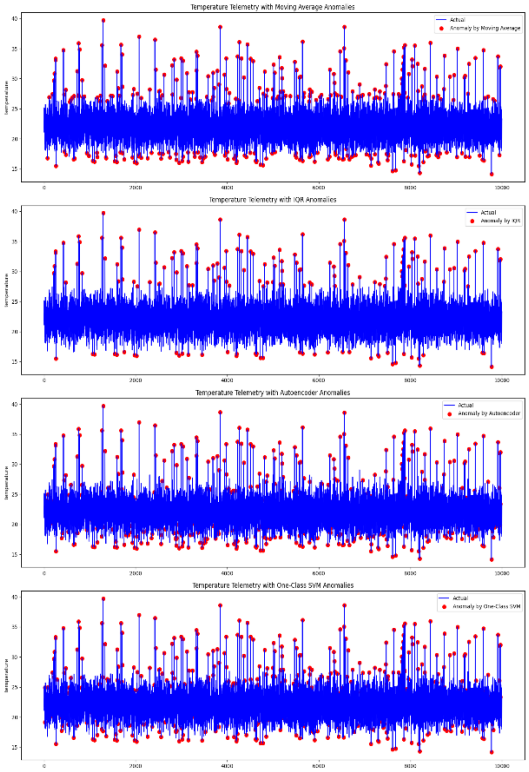

Figure 6. Comparison of performance of All the Methods Using This Study.

4. Conclusion

The culmination of our study on anomaly detection in synthetic IoT telemetry data brings forth several significant findings and insights. Through a meticulous analysis of various anomaly detection methods, we have unearthed essential information about their performances and applicability in specific contexts.

Our investigation revealed the distinct superiority of Autoencoders in detecting anomalies within the synthetic dataset. Their ability to learn complex data patterns and identify subtle deviations highlights the potential of deep learning techniques in sophisticated anomaly detection tasks. Autoencoders demonstrated a high degree of accuracy, as evidenced by their F1-score of 0.72, making them particularly suitable for complex IoT environments where anomalies may not be overtly pronounced.

In contrast, traditional methods like the Moving Average and Interquartile Range (IQR) methods, though outperformed by Autoencoders, still hold significance due to their simplicity and ease of implementation. These methods, with their lower computational demands, are apt for scenarios where rapid, less resource-intensive anomaly detection is required. Their effectiveness in identifying more pronounced anomalies makes them valuable as preliminary screening tools in a multi-tiered detection strategy.

The One-Class SVM, another key method explored in this study, also demonstrated commendable performance, with an F1-score of 0.46. This method's ability to create decision boundaries in high-dimensional spaces makes it a worthy contender in the field, especially in handling complex data structures where traditional statistical methods may falter.

From a practical standpoint, the implications of these findings for IoT device monitoring are profound. As IoT devices continue to proliferate and generate vast quantities of data, the need for efficient and accurate anomaly detection systems becomes increasingly crucial. Our study suggests that a hybrid approach, employing both traditional and advanced techniques like Autoencoders, might offer the most comprehensive solution for real-world applications. Such a combination could leverage the rapid detection capabilities of traditional methods and the nuanced analysis of deep learning models, ensuring thorough and efficient monitoring of IoT devices.

Looking forward, there are several avenues for future research in this domain. One promising area is the exploration of hybrid models that combine the strengths of different anomaly detection techniques. Additionally, investigating the scalability of these methods and their applicability in real-time IoT environments presents another fruitful area of research. Further studies could also delve into the optimization of deep learning models for anomaly detection, focusing on reducing their computational requirements while maintaining high accuracy.

In conclusion, our study provides valuable insights into the performance of various anomaly detection methods in the context of IoT telemetry data. It underscores the importance of selecting appropriate methods based on specific data characteristics and highlights the potential of deep learning techniques in handling complex anomaly detection tasks. As the IoT landscape continues to evolve, these findings will play a crucial role in shaping effective and efficient anomaly detection strategies for IoT device monitoring.

Acknowledgments

Reviewers and the editor's comments have been greatly appreciated by the author. Additionally, I appreciate Dr. Mohammad Amin Khalili's assistance in developing and implementing deep learning algorithms.

Refrences

[1] S. M. A. Group et al., “Internet of Things (IoT): A Literature Review,” J. Comput. Commun., vol. 03, no. 05, Art. no. 05, 2015, doi: 10.4236/jcc.2015.35021.

[2] J. A. Marques, M. C. Luizelli, R. I. Tavares da Costa Filho, and L. P. Gaspary, “An optimization-based approach for efficient network monitoring using in-band network telemetry,” J. Internet Serv. Appl., vol. 10, no. 1, p. 12, Jun. 2019, doi: 10.1186/s13174-019-0112-0.

[3] S. Clements, D. Jepsen, M. Karnowski, and C. B. Schreck, “Optimization of an Acoustic Telemetry Array for Detecting Transmitter-Implanted Fish,” North Am. J. Fish. Manag., vol. 25, no. 2, pp. 429–436, May 2005, doi: 10.1577/M03-224.1.

[4] E. Benkhelifa, T. Welsh, and W. Hamouda, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems," IEEE Commun. Surv. Tutor., vol. 20, no. 4, pp. 3496–3509, 2018, doi: 10.1109/COMST.2018.2844742.

[5] M. Fahim and A. Sillitti, "Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review," IEEE Access, vol. 7, pp. 81664–81681, 2019, doi: 10.1109/ACCESS.2019.2921912.

[6] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Comput. Netw., vol. 51, no. 12, pp. 3448–3470, Aug. 2007, doi: 10.1016/j.comnet.2007.02.001.

[7] M. Stolpe, "The Internet of Things: Opportunities and Challenges for Distributed Data Analysis," ACM SIGKDD Explor. Newsl., vol. 18, no. 1, pp. 15–34, Aug. 2016, doi: 10.1145/2980765.2980768.

[8] A. Lavin and S. Ahmad, "Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark," in 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Dec. 2015, pp. 38–44. doi: 10.1109/ICMLA.2015.141.

[9] A. B. Ashfaq, M. Q. Ali, and S. A. Khayam, "Accuracy improving guidelines for network anomaly detection systems," J. Comput. Virol., vol. 7, no. 1, pp. 63–81, Feb. 2011, doi: 10.1007/s11416-009-0133-5.

[10] M. E. Civelek and O. K. Artar, "Blockchain and Artificial Intelligence Technologies for Balanced Foreign Trade: Replacing Exchange Function of Money." Rochester, NY, Aug. 31, 2019. Accessed: Mar. 15, 2023. [Online]. Available: https://papers.ssrn.com/abstract=3450291

[11] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in Nonstationary Environments: A Survey," IEEE Comput. Intell. Mag., vol. 10, no. 4, pp. 12–25, Nov. 2015, doi: 10.1109/MCI.2015.2471196.

[12] S. Hansun, "A new approach of moving average method in time series analysis," in 2013 Conference on New Media Studies (CoNMedia), Nov. 2013, pp. 1–4. doi: 10.1109/CoNMedia.2013.6708545.

[13] "The interquartile range: Theory and estimation - ProQuest." Accessed: Nov. 21, 2023. [Online]. Available: https://www.proquest.com/openview/8449e263bd9f96a22e0348e6abdeb5a9/1?pq-origsite=gscholar&cbl=18750&diss=y

[14] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," IEEE Internet Things J., vol. 7, no. 7, pp. 6481–6494, Jul. 2020, doi: 10.1109/JIOT.2019.2958185.

[15] F. Palmieri and U. Fiore, "Network anomaly detection through nonlinear analysis," Comput. Secur., vol. 29, no. 7, pp. 737–755, Oct. 2010, doi: 10.1016/j.cose.2010.05.002.

[16] L. Erhan et al., "Smart anomaly detection in sensor systems: A multi-perspective review," Inf. Fusion, vol. 67, pp. 64–79, Mar. 2021, doi: 10.1016/j.inffus.2020.10.001.

[17] L. Fernández Maimó, Á. L. Perales Gómez, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez, "A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks," IEEE Access, vol. 6, pp. 7700–7712, 2018, doi: 10.1109/ACCESS.2018.2803446.

[18] W. H. Lopez Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Chapter 11 - Autoencoders," in Machine Learning, A. Mechelli and S. Vieira, Eds., Academic Press, 2020, pp. 193–208. doi: 10.1016/B978-0-12-815739-8.00011-0.

[19] K. Sun, J. Zhang, C. Zhang, and J. Hu, "Generalized extreme learning machine autoencoder and a new deep neural network," Neurocomputing, vol. 230, pp. 374–381, Mar. 2017, doi: 10.1016/j.neucom.2016.12.027.

[20] D. Y. Oh and I. D. Yun, "Residual Error Based Anomaly Detection Using Auto-Encoder in SMD Machine Sound," Sensors, vol. 18, no. 5, Art. no. 5, May 2018, doi: 10.3390/s18051308.

[21] L. Feeken et al., "Detecting and Processing Anomalies in a Factory of the Future," Appl. Sci., vol. 12, no. 16, Art. no. 16, Jan. 2022, doi: 10.3390/app12168181.

[22] G. Zachos, I. Essop, G. Mantas, K. Porfyrakis, J. C. Ribeiro, and J. Rodriguez, "Generating IoT Edge Network Datasets based on the TON_IoT Telemetry Dataset," in 2021 IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Oct. 2021, pp. 1–6. doi: 10.1109/CAMAD52502.2021.9617799.

[23] P. M. S. Sánchez, J. M. J. Valero, A. H. Celdrán, G. Bovet, M. G. Pérez, and G. M. Pérez, "A Survey on Device Behavior Fingerprinting: Data Sources, Techniques, Application Scenarios, and Datasets," IEEE Commun. Surv. Tutor., vol. 23, no. 2, pp. 1048–1077, 2021, doi: 10.1109/COMST.2021.3064259.

[24] M. L. Ravalec, B. Noetinger, and L. Y. Hu, "The FFT Moving Average (FFT-MA) Generator: An Efficient Numerical Method for Generating and Conditioning Gaussian Simulations," Math. Geol., vol. 32, no. 6, pp. 701–723, Aug. 2000, doi: 10.1023/A:1007542406333.

[25] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Comput. Secur., vol. 28, no. 1, pp. 18–28, Feb. 2009, doi: 10.1016/j.cose.2008.08.003.

[26] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An ensemble deep learning model for cyber threat hunting in industrial internet of things," Digit. Commun. Netw., vol. 9, no. 1, pp. 101–110, Feb. 2023, doi: 10.1016/j.dcan.2022.09.008.

[27] I. Kiss, B. Genge, and P. Haller, "A clustering-based approach to detect cyber attacks in process control systems," in 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Jul. 2015, pp. 142–148. doi: 10.1109/INDIN.2015.7281725.

[28] P. Mansourian, N. Zhang, A. Jaekel, and M. Kneppers, "Deep Learning-Based Anomaly Detection for Connected Autonomous Vehicles Using Spatiotemporal Information," IEEE Trans. Intell. Transp. Syst., pp. 1–12, 2023, doi: 10.1109/TITS.2023.3286611.

[29] C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak, "XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification," Anal. Chem., vol. 78, no. 3, pp. 779–787, Feb. 2006, doi: 10.1021/ac051437y.

[30] H. H. van Rossum, "Moving average quality control: principles, practical application and future perspectives," Clin. Chem. Lab. Med. CCLM, vol. 57, no. 6, pp. 773–782, Jun. 2019, doi: 10.1515/cclm-2018-0795.

[31] H. P. Vinutha, B. Poornima, and B. M. Sagar, "Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset," in Information and Decision Sciences, S. C. Satapathy, J. M. R. S. Tavares, V. Bhateja, and J. R. Mohanty, Eds., in Advances in Intelligent Systems and Computing. Singapore: Springer, 2018, pp. 511–518. doi: 10.1007/978-981-10-7563-6_53.

[32] M. Sewak, S. K. Sahay, and H. Rathore, "An Overview of Deep Learning Architecture of Deep Neural Networks and Autoencoders," J. Comput. Theor. Nanosci., vol. 17, no. 1, pp. 182–188, Jan. 2020, doi: 10.1166/jctn.2020.8648.

[33] J. Mũnoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls, "Semisupervised One-Class Support Vector Machines for Classification of Remote Sensing Data," IEEE Trans. Geosci. Remote Sens., vol. 48, no. 8, pp. 3188–3197, Aug. 2010, doi: 10.1109/TGRS.2010.2045764.

[34] M. A. Khalili, B. Voosoghi, L. Guerriero, S. Haji-Aghajany, D. Calcaterra, and D. Di Martire, "Mapping of Mean Deformation Rates Based on APS-Corrected InSAR Data Using Unsupervised Clustering Algorithms," Remote Sens., vol. 15, no. 2, Art. no. 2, Jan. 2023, doi: 10.3390/rs15020529.

[35] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, p. 6, Jan. 2020, doi: 10.1186/s12864-019-6413-7.

[36] M. A. Khalili, L. Guerriero, M. Pouralizadeh, D. Calcaterra, and D. Di Martire, "PREDICTION OF DEFORMATION CAUSED BY LANDSLIDES BASED ON GRAPH CONVOLUTION NETWORKS ALGORITHM AND DINSAR TECHNIQUE," ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci., vol. X-4-W1-2022, pp. 391–397, Jan. 2023, doi: 10.5194/isprs-annals-X-4-W1-2022-391-2023.

[37] M. A. Khalili, L. Guerriero, M. Pouralizadeh, D. Calcaterra, and D. Di Martire, "Monitoring and prediction of landslide-related deformation based on the GCN-LSTM algorithm and SAR imagery," Nat. Hazards, Aug. 2023, doi: 10.1007/s11069-023-06121-8.