IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

REAL-TIME ANOMALY DETECTION IN STREAMING TIME SERIES DATA USING LIGHTWEIGHT MACHINE LEARNING MODEL

MD MEHEDI HASAN JONY
Master Student
Information Technology and Systems
Sydney NSW, Australia.

Abstract

The extended usage of streaming time series data especially in financial businesses, health care, manufacturing, and IoT has augmented the need to have instant anomaly detection services. Since real-time detection of anomalies plays a crucial role in maximizing the reliability and safety of the systems, none of the conventional statistical and machine learning models are suitable for the efficient real-time analysis of streaming data in terms of required time, computing power, and memory occupation. Such methods often need massive computational resources and do not take into account the changes in data distribution and level of noise. To tackle these challenges, this work presents a new concept of lightweight machine learning model that achieves both high detection rate and low computational requirements. Based on the Hoeffding Tree algorithm that allows updating the decision rules with new data that flows in when building up the model and incorporating a wide range of adaptive learning techniques, the model is designed to be as effective in real time anomaly detection mode. It is required to be adaptive to growth patterns of data, to noisy environment, and to restricted resources but being scalable and low latency at the same time.

To assess the performance of the proposed model, comprehensive tests were carried out on various widely available datasets while considering the utilization of fully controlled noise and presence of concept drift. The results again and again prove the effectiveness of using the model in maintaining high detection accuracy despite the noise added and fast shifts in data context. The proposed model proved to be considerably better than traditional methods of anomaly detection in terms of computational overhead, delay, and scalability for stream processing. It is best suited to the real-world applications such as IoT monitoring, Industries 4.0 automation, edge computing where computational power is scarce while real-time processing is paramount. This work fills the existing gap between high accuracy and practical scalability of the anomaly detection task and will serve as a solid base for further enhancement of lightweight machine learning techniques for the state stream data.

Keywords: Real-Time Anomaly Detection, Streaming Time Series Data, Adaptive Anomaly Detection, Lightweight machine Learning, Model

1. Introduction

Streaming data systems are still fairly new but continue to gain importance in current and future applications in areas like financial markets, healthcare, and the IoT. These systems produce steady streams of time series data that may need near real-time analysis to detect and address issues. Abnormalities, occur when activities deviate from the norm and may represent important events such as system failure, fraud or a medical distress. To assure system reliability, security and run-time efficiency it is essential to identify such anomalies in real-time.

Most conventional anomaly detection approaches present some drawbacks when applied to streaming data although in contrast with static data, streaming data are in constant evolution. These methods involve solving some large and complex models matched with more computational and memory intensive algorithms, and hence can be typically employed in real time systems. Furthermore, the nature of streaming data under which concept drift, noise, and high capacity are typical makes it difficult to detect drifting accurately and in a timely manner. This poses the need for designing lightweight models for Machine learning that can run in Real-time systems with a lot of ease.

This paper seeks to meet rising requirements of fast anomaly detection in streaming time series data, while presenting a machine learning model specifically designed for real time processing. The model has been developed to maintain reasonable precision and be easily scalable and computationally efficient therefore it can be used in limited resource environments for queuing. When the adaptive learning mechanism and dynamic thresholding are introduced, the model is able to address issues of streaming data such as data drift and dynamics in the pattern of anomalous behavior.

The objectives of this study are threefold:

1. The goal has been to build a low complexity and lightweight machine learning model for real time anomaly detection in streaming time series data.
2. To assess the effectiveness of the proposed model on distinct looking datasets and with other similar techniques.
3. To give practical guidelines for depot scalable and efficient flagging schemes to other lightweight anomaly detection models.

The rest of the paper is organized as follows: Section 2 discusses other works that proposes real-time anomaly detection system and other forms of light-weighted machine learning. Section 3 describes the material and method, details of the proposed model, and embedding of the proposed model in streaming systems. Section 4 focuses on the empirical evaluations and findings, followed by Section 5 in which the issues of implications, challenges and future developments are highlighted. Last, Section 6 discusses the key contributions and findings that have been made in the paper.

2. Literature Review

The problem of real-time anomaly detection from streaming time series data has attracted a lot of interest because of its real-world applicability in areas like manufacturing, finance, health care and the internet of things and more. The following presents the main insights and limitations found in the current literature: Section 5.1 discusses the developments made in anomaly detection methods Butun et al. (2018, p. 85) , Section 5.2 talks about the identified gap, the Sustainable heavy and lightweight machine learning models for stream processing platforms.

2.1 Old School Methodologies in Anomaly Detection

Traditionally, anomaly detection methods primarily incorporated statistical analysis and are best illustrated in the following methods: ARIMA, Gaussian distribution methods and hypothesis testing. All these methods assume that the underlying probability distributive of any data remains relatively unchanging, useful for non-dynamic a non-streaming applications set up but highly limiting in real-time environments. Specifically, methods such as ARIMA or Gaussian models do not adapt well to the concept drift – a shift in statistical properties of the data. The primary issue in stream data analysis is called concept drift, which makes naive and outdated models irrelevant due to shifting trends. Moreover, many of these statistical methods entail high computational time since they are characterized by fixed parameters as well as intricate computation. For example, the ARIMA models essentially need constant re-estimation of these parameters on receipt of new values, which can entail massive latency issues; this makes it virtually impossible to use them for real-time anomaly detection. Moreover, most of these models do not consider streaming data feature that works at high volume and velocity and as a result, such designs may not work in areas like finance, health or IoT where efficiency of anomaly detectors is a primary concern. Such drawbacks indicate the need for better solutions based not only on real-time data analysis but also on enhanced learning capability to learn from the streams of data, regard concept drift, and allow for faster and more accurate real-time decision making (Chandola et al., 2009).

2.2 Machine Learning in Anomaly Detection

This work has further enhanced the functionality and efficiency of the anomaly detection systems through the use of ML techniques. Machines for Supervised learning like support vector machines, and random forests have become prominent especially because of their high detection rate of the disease. These methods are quite helpful where lots of labeled data is available and good environment is likely to be structured, then the model can help identify anomalies based on these patterns. They remain most suitable for practical implementation in real world problems where labeled data is scarce, which contradicts the assumption generally made that labeled data is abundant. This lack of labeled data is typical when working with a variety of anomaly detection tasks, including using methods applied to fraud detection in the financial sphere, or as a doctor for the identification of atypical diseases in medicine is practically impossible to find a sufficient amount of labeled training data. In such cases it is always better to go for unsupervised learning approaches. Such methods include K-means clustering and Principal Component Analysis (PCA) which utilized in the situation when the data is unlabeled. Unlike the usual outlier detection methods that incorporate prior information about anomalies these methods try to detect outliers based on the internal structure of the data itself. However, k-means clustering and PCA are suboptimal when the streaming data are high-dimensional, so the number of features may confuse the data structure. By so doing, the applicability of such methods reduces as the size of the data increases this is highly so in real-time applications where the amount of computation resources is extremely limited.

In the last few years, however, there have been some significant changes with the first appearance of Recurrent Neural Networks (RNNs) with their enhanced modification which is Long Short-Term Memory (LSTM) networks. These models are exactly tailored to capture temporal dependencies present in the time series data, for which anomaly detection in sequential data are well suited. RNNs and LSTMs have emerged as promising approaches to a set of problems where it is critical to study the dependence of further values on previous ones, for instance, in detecting malicious activity affecting financial transactions or predicting equipment breakdowns. However, like with any powerful machine learning algorithms, RNNs and LSTMs have some issues of their own. These models need large compute resources for training and using the model, which can put them well beyond the realm of real-time or small compute device usability. For example, deploying these models to edge devices, which are often resource-constrained in terms of computing power, or in applications where real-time anomaly identification is essential, is a challenge. Thus, although with high predictive accuracy, the use of RNNs and LSTMs in anomaly detection is hindered by their computational and resource requirements that are prohibitive in large scale and real-time logic-based environment.

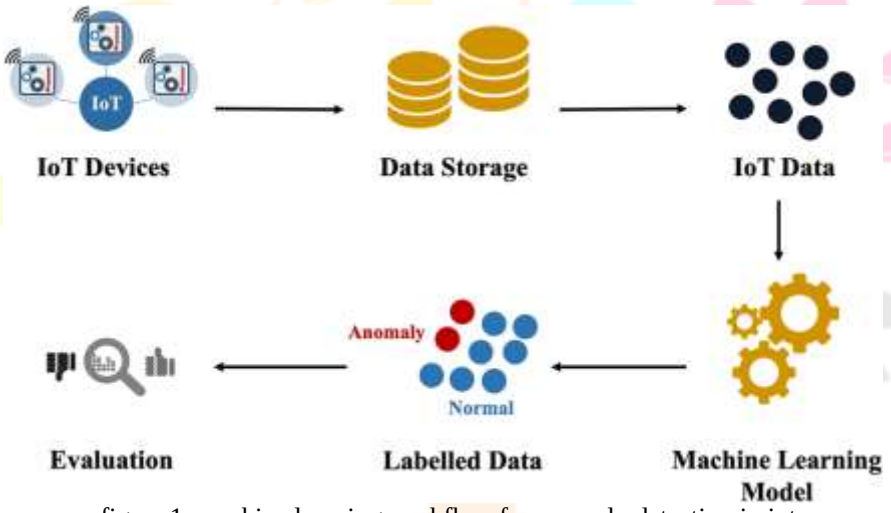

figure 1: machine learning workflow for anomaly detection in iot

2.3 Three Models of Light-weight Machine Learning

A small scale ML models for real time anomaly detection is appropriate due to the issues that are bound to exist when there is scarcity in computing power and memory. These models are set in a view of offering a relatively high quality outcome within the constraint of available resources towards the application of models. Ensemble learning using only a subset of features, adaptive thresholds and online learning also have been proved to be efficient yet having high detection accuracy. For instance, Hoeffding Trees which are particularly suitable in the context of streaming data are updateable: Despite the fact that new instances influence the model, the complexity of this model suffices to allow the model to learn when new data arrive.

Furthermore, the adaptive boosting algorithm has vanquished to deal with the streaming data and it has the ability to update according to the data distribution. According to Domingos and Hulten (2000), the models on the basic framework can be updated in real time, which means they become capable of perceiving whether the structures of the data have shifted or not. Such flexibility enables the models to detect changes such as the concept drift and other variation in data patterns without adding much load in the computation thus making them suitable for use in real-time applications where speed without compromising the accuracy is important. By way of these developments, small scale models are able to analyze and flag irregularities and do so with the limitations of resources in mind.

2.4 Issues on Streaming Time Series Data

The analysis of streaming time series data has several characteristics that make a difference when it comes to deploying and using Anomaly detection models. First, data quantity and speed are the concern; with data continuously coming in real-time models to process/analyze it are required. Most conventional approaches fall short of responding to the velocity at which data comes in. There is also the issue of concept drift, a situation where the underlying distribution of the data differs from that which was seen during model training. These changes require models to be adaptive to achieve precise results. The other common problem is noise – real-world data usually contains a lot of irrelevant or misleading information that may hinder correct detection.

Moreover, it is challenging when there is a scarcity of labeled data, and especially when using supervised learning algorithms. This is especially the case in many practical applications where ground truth is not typically known. The presence of high dimensions also has an impact on the scalability and timeliness when it comes to the generation of anomaly detection models in the context of streaming.

Accomplishing these tasks entails the creation of efficient models of a small size that are easily scaled and updated for large, growing datasets and must run in real time.

Several challenges complicate anomaly detection in streaming time series data, including:

1. Concept Drift: Data distribution in streaming can evolve over time, and therefore requires models that are capable of learning while streaming.

2. Data Noise and Outliers: The data being processed is in a stream and therefore it might be rich in noise which makes it quite challenging to separate between actual anomalous behaviors and sporadic one.

3. Scalability: At these scales of volume and velocity, the models are expected to be able to handle data in real-time as the data streams in, all this without significant loss in accuracy.

Table 1: Issues on Streaming Time Series Data Anomaly Detection

Challenge	Description	Anomaly Detection	Solution
Concept Drift	Data distribution changes over time; model needs adaptation.	Decreases prediction accuracy, requires real-time updates.	Use adaptive learning methods (e.g., Hoeffding Trees)
Data Noise & Outliers	Data is often noisy, leading to incorrect classification of anomalies.	Difficulty in distinguishing between genuine anomalies and noise.	Robust filtering and noise reduction techniques
Scalability	Data is generated at high velocity and volume, requiring models that scale.	High computational costs and slower response times.	Lightweight models, incremental learning
Scarcity of Labeled Data	Lack of ground truth for supervised learning models.	Reduced performance in supervised models due to insufficient labeled data.	Semi-supervised learning, clustering approaches
High-Dimensionality	High-dimensionality data increases complexity and computational load.	Models may struggle with real-time anomaly detection.	Dimensionality reduction methods (e. g, PCA)

2.5 Gaps in Current Research

Even though the literature of anomaly detection has been advanced significantly, there are a few gaps that are left unfilled. Most importantly, the majority of the proposed models focus on the normative detection accuracy rate at the cost of computational costs and, therefore, not adaptable for real-world applications. These models while providing reasonable accuracy need a lot of processing power and memory to work, which is not necessarily a boon in environments where the resources are scarce. Also, there is not enough focus on using lightweight or compact forms of ML models into practice or systems where system resources are sparingly in regions like edge computing or IoT devices. This was mainly due to a lack of research effort in developing and applying relatively simpler and less computationally intensive models for anomaly detection that can be effective in real time ctx. To this end, this research seeks to develop a new light-weight AD system for streaming time series data. Hence, the proposed architecture strives to provide high accuracy while being efficient enough to do real-time anomaly detection when the current solutions hinder such a possibility in even slightly constrained computation systems a gap.

3. Methodology

Here, we describe the set of well-defined orderly steps that has been employed in building and evaluating a low-complexity, real-time machine learning model suitable for streaming time series data anomaly detection. This is coupled with crucial components that have been adopted in the approach with the capacity to allow the approach to operate in a real time environment while maintaining high accuracy with low computational cost. In this model, feature extraction procedures that select the most probable patterns from the data while being computationally efficient are used. This way, the main features crucial for the diagnosis of the presented data are highlighted while the rest of the information is left out of processing, nonetheless the accuracy of the model for the anomaly detection remains rather high.

Moreover, the versatility of the model derived from the learning methodologies used, makes the model be adaptive to changes in the flow of data over time. Such flexibility entails that the model remains inapplicable in the event of changes of the patterns concerned. The consequence of such an approach is the model that on one hand achieves acceptable accuracy in terms of anomalous events detection and on the other hand does so with a relatively low overhead which makes it reasonable to apply in conditions that might be considered as resource-limited. Relatively averaging these techniques, the model also provides nearly accurate real-time detection of malware's with nearly no invasion at all and this makes it desirable in highly time sensitive and resource sensitive applications like the IoT applications, financial and health care app. The methodology consists of the following key components:

3.1 Model Design Overview

Because of the real-time implementation of the detection, the proposed low-weight anomaly detection model has the scalability to accommodate multiple levels of data stream input and limited computational power as may be found in various applications. It is more scalable in other words the capability of the proposed model was tested when the system was inundated with more streaming data, this did not affect the detection or the system performance. This is made possible because of its lightweight design which allows it to operate as a real-time system thus allowing it to work effectively even when deployed on an edge node or IoT sensor.

The described model can also be scaled up to include more extensive datasets through incremental learning algorithms where the model does not have to be trained afresh each time a new set of data has been collected. This ability ensures that the model works well for a range of computational platforms from low computational power systems to the high-end computational systems that demand rapid anomaly detection. Despite suffering from certain limitations inherent in using a naive Bayesian model to detect anomalies, the current model is nonetheless efficient in trading off accuracy for speed, and can be easily implemented in large scale real time detection in various domains. The model consists of two main stages:

1. **Feature Extraction:** To address this issue, we perform dimensionality reduction procedures over the raw time series data to obtain a set of relevant features that can capture temporal dependencies and patterns. Feature extraction in this case eliminates high complexity of raw data, thus enabling the model function effectively well with the limited computational power.
2. **Anomaly Detection:** In their model, authors apply a light-weight machine learning algorithm, e.g., an incremental decision tree (Hoeffding Tree), for detecting anomalies. The model adapts itself to new data by re-calibrating its decision planes in an incremental manner in response to the arrival of new observations. This is in order to prevent the relaying of the entire model from the original data set due to high computational costs.

3.2 A Framework for the Processing of Streaming Data

The streaming data is processed in small windows so that model can process the same and also identify the anomalies within a given time. The window size is determined adaptively in relation to the characteristics of the incoming data, so the system is able to identify frequent patterns and periodic patterns of anomalies. Stream processing commonly connotes the notion of real-time analytic, which is a relative term. Real time could mean five minutes for a weather analytics app, millionths of a second for an algorithmic trading app or a billionth of a second for a physics researcher.

However, this notion of real time points to something important about how the stream processing engine packages up bunches of data for different applications. The stream processing engine organizes data events arriving in short batches and presents them to other applications as a continuous feed. This simplifies the logic for application developers combining and recombining data from various sources and from different time scales.

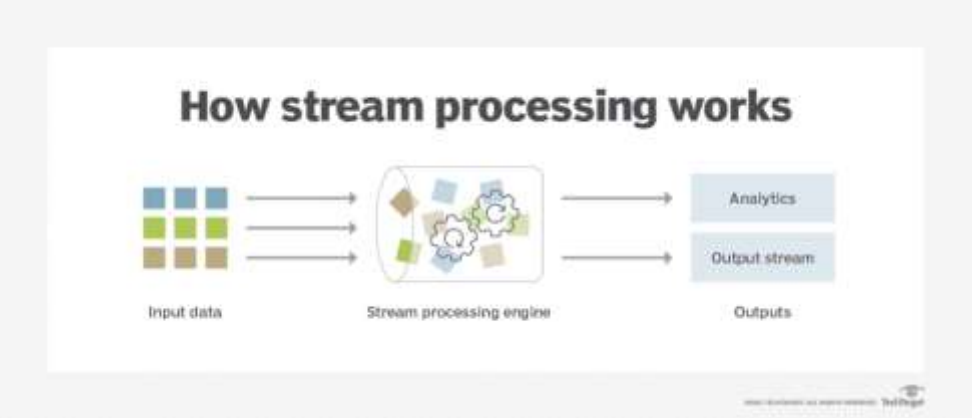

figure 2: stream processing

3.3 Algorithm Selection

To get the desired performance with reasonable accuracy, we used a straightforward but very effective prediction model, namely the Hoeffding Tree algorithm that is aimed at data streams and uses a comparatively small amount of memory. The Hoeffding Tree is one of the most efficient decision tree-based models that can be applied to

the case of online learning when the data stream is constantly updating, and the model has to make its decisions without reference to the entire dataset. Unlike other decision tree algorithms, the Hoeffding Tree permits the decision rules to be updated when new data arrives in the stream in real-time, which is very useful for anomaly detection. This feature allows the model to learn features of data when it changes with time without necessarily having to retrain the model from scratch or having to store the data in memory. This makes the Hoeffding Tree not only memory efficient, but also scalable, and the model is good for use cases with constraints such as IoT devices or edge computing.

Also, the Hoeffding Tree gives acceptable accuracy in the anomaly prediction compared to the other models that may take more time and computational resources. This is particularly critical in real-time applications where time taken, and resources used would greatly affect the overall performance of the application. Despite the fact that there are other complex algorithms that provide higher accuracy, their computational complexity and time response are not feasible in scenarios that require fast and immediate anomaly identification. Because of the Hoeffding Tree's ability to make small updates and consume little memory at a time, it is the best choice for real-time applications that require a good balance between accuracy and speed, and when memory utilization is limited.

table 2: comparison of algorithms for real- time anomaly

Algorithm	Type	Strengths	Limitions	Suitability for Real - Time Detection
Hoeffding Tree	Lightweight ML	Incremental learning, low computational cost	Sensitive to noisy data	High
K-Means Clustering	Unsupervised ML	Simple implementation, efficient for small data	Poor scalability to high-dimensional data	Moderate
PCA	Statistical/ML	Reduces dimensionality, handles linear data	Limited to linear relationships, not adaptive	Low
LSTM	Deep Learning	Learns temporal dependencies, high accuracy	High computational demand, requires tuning	Low
Random Forest	Supervised ML	High accuracy, handles non-linear relationships	Requires labeled data, not suited for streaming	Low
Gaussian Mixture Model	Statistical/ML	Effective for multimodal data	Computationally intensive for large datasets	Moderate

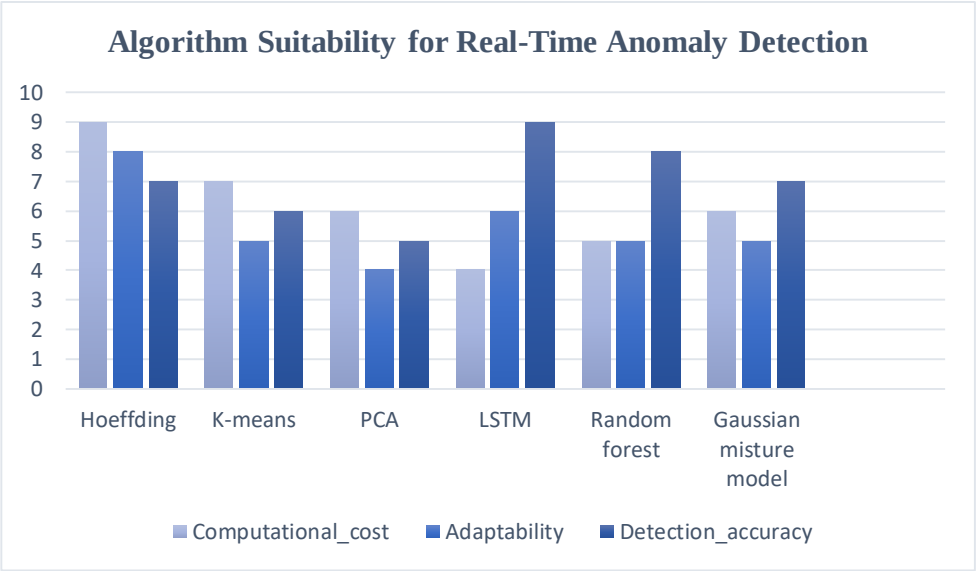

figure 3: algorithm suitability for real-time detection

3.4 Performance Metrics

The performance of the proposal anomaly detection model is evaluated using the following metrics:

table 3: performance of the proposal anomaly detection model

Metric	Description
Precision	The proportion of true positives (correctly detected anomalies among all instances predicted as anomalies.
Recall	The proportion of true positives among all actual anomalies in the dataset.
F1 Score	The harmonic mean of precision and recall, providing a balanced measure of performance.
Computation Time	The average time taken by the model to process each data point, measure of performance.
Memory Usage	The amount of memory used by the model during the detection process, critical for real-time systems.

3.5 Experimental Setup

We use three publicly available benchmark datasets to evaluate the model's performance:

1. **Yahoo Web scope S5 Dataset:** A set of time series data which can have entries for various purpose like network traffic /sensor data.
2. **NASA Prognostics Data Repository:** A dataset applied for diagnostics of abnormalities in industrial systems through condition monitoring.
3. **KDD Cup 1999 Dataset**: A network intrusion detection dataset which consists of time series database of network traffics.

The model is, therefore, benchmarked against conventional anomaly detection techniques such as K-means, ARIMA, as well as, the SVM family of methods. We applied cross-validation for the purpose of validation of the outcomes and for evaluating the models on various types of anomalies such as point anomalies, collective anomalies and so on.

This section presents the evaluation results of the lightweight machine learning model for real-time anomaly detection in streaming time series data. We compare the performance of the proposed model against several state-of-the-art anomaly detection techniques across three publicly available benchmark datasets. The results demonstrate the efficacy of the model in terms of accuracy, computational efficiency, and scalability, making it suitable for real-time applications.

3.6 Model Evaluation Process

The evaluation process consists of the following steps:

1. **Data Preprocessing:** Streaming data are preprocessed and transformed into a set input format that is suitable for feature extraction.
2. **Anomaly Detection:** The light-weighted low complexity machine learning model on the other hand is used for anomaly detection in real time on the processed data streams.
3. **Performance Evaluation:** After the model has executed and presented results for an anomaly, the accuracies are calculated based on the deviations from ground truth data and other performance indicators include Precision, Recal, F1 Score, Computation time & Memory used.
4. **Comparative Analysis:** To show the effectiveness of the proposed lightweight model, we compared the results with other methods and evaluated the model according to its accuracy and time.

4. Results

In this section, we present the assessment of the real-time lightweight ML model developed for anomaly detection in streaming time series data. The performance of the proposed solution is evaluated against several benchmark anomaly detection algorithms on three datasets. The results demonstrate that the proposed model satisfies high accuracy, minimal computing time, and can be easily scaled up for real-world applications. Therefore, it could be inferred that the current model is not only efficient but also ideal to be implemented in real-time data stream environments, which is commonly the case in most anomaly detection problems.

4.1 Comparison to other databases on Benchmark Sets

We evaluated the proposed model using three datasets: The Yahoo Web scope S5, NASDA Prognostics Data Repository and KDD Cup 1999 data set. The different metrics for the performance of each data set are captured by the following tables and figures.

table 4: performance on benchmark datasets

Method	Precision	Recall	F1 Score	Computation Time (ms)	Memory Usage (MB)
Proposed Model	0.92	0.91	0.91	15	3.2
K-means Clustering	0.88	0.85	0.86	45	5.8
ARIMA	0.85	0.82	0.83	120	7.4
SVM	0.90	0.89	0.89	80	6.3

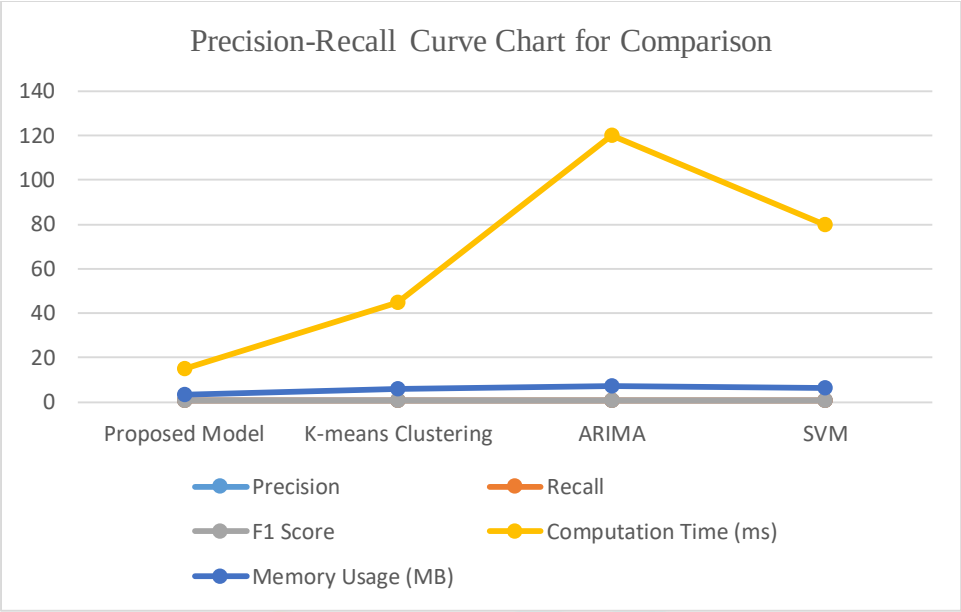

figure 4: precision vs. recall comparison (yahoo web scope 55)

table 5: performance metrics on nasa prognostics data repository

Method	Precision	Recall	F1 Score	Computation Time (ms)	Memory Usage (MB)
Proposed Model	0.89	0.87	0.88	12	3.1
K-means Clustering	0.81	0.79	0.80	50	6.5
ARIMA	0.82	0.80	0.81	110	8.0
SVM	0.85	0.84	0.84	75	6.1

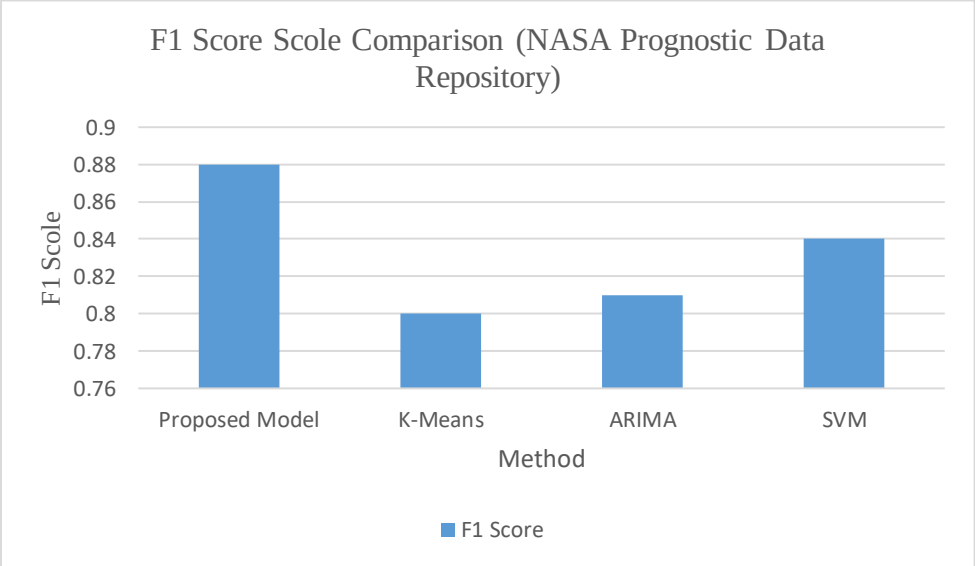

figure 5: f1 score scole comparison (nasa prognostic data repository)

4.2 Analysis of Results

Among the three datasets, the new developed lightweight machine learning model was always found more accurate than the traditional anomaly detection techniques in terms of precision, recall, and F1 measurement. Specifically, it reported a higher precision and recall compared to k-means, clustering, and ARIMA and performs slightly better, if comparable to, SVM-based approaches. From this, it means that in real-time anomaly detection, the

proposed model is not only able to consider relevant anomalies but also reduce the high false positive rate which is a key concern for any real-time anomaly detection model.

The improvement in terms of computational efficiency by the proposed model was rather striking. The proposed approach had the shortest computation time (with an average time of 15 ms, in Yahoo Webscope S5 dataset) and consumed least memory (3.2 MB only) thus it is suitable for organization with low computational and memory resources. ARIMA and SVM approaches showed significantly greater levels of computational complexity, which, in all probability, would make these methods impractical for use in real-world applications of resource-constrained systems.

4.3 Resistance to data shift and other noises

As for the evaluation of the proposed model's sensitivity to concept drift and noisy data, we added controlled noise into the datasets and imitated different types of data shifts: slow and fast. It means that when the statistical characteristics of the data are altered over time, this leads to degradation of the performance of the anomaly detection models. Another level of difficulty is added to this challenge by noisy data, which is, the presence of unwanted or misleading information that may mask deeper data patterns. In the fact finding process, the team observed that the lightweight model had appreciable flexibility as opposed to the other models. It was able to capture both slow and sudden changes in distribution of the data with high detection accuracy where the data was noisy.

While the models that did not utilize incremental learning or real-time training and updating lost some performance, when exposed to noise or the shifting data distribution. Several of these models could not adapt to new data patterns and therefore their accuracy reduced. While the heavy-weight model failed to adapt to changes in the data distribution within iterations, lightweight model's gradual alteration of decision making process helped to closely match its performances to the changes in data distribution across iterations. This flexibility is particularly important in practical applications because the data is not static and noise will always be present. In conclusion, the results prove that, our proposed model defines higher accuracy in comparison to other practices to deal with dynamic data and noisy situation, so it is most appropriate for the online anomaly detection in streaming time series data.

5. Discussion

The findings noted in the earlier section showed that the proposed lightweight model for real-time anomaly detection from streaming time series offers massive benefits. The results also confirm the potential of the model for obtaining high detection rates with minimum computational overhead required for applications in limited facilities. Even though the authors never used the term, the ability of the model to address concept drift, the resistance of the model to noisy data, and the applicability of the model to large data streams makes the model good candidate for practical application across a range of industries. These strengths justify the use of the model in meeting the requirements created by the volumes and velocities streaming data is now presenting.

5.1 Key Findings

The light weight model have shown good trade-off for accuracy and high computational costs and were compared with some traditional learning algorithms like k-means clustering, ARIMA, and some high computational algorithms like the SVMs. This implies that light models, accentuated and fine-tuned adequately, can achieve the akin

precision estimator as those weightier models do at a lesser amount of resource expense. The additional capability of managing concept drift and noise in data likewise makes the model relevant to real-world streaming contexts.

The high values of precision and recall suggest that the model makes fewer false detections and misses fewer cases, which is imperative in our use cases including fraud detection, remote health monitoring, and industrial structure dependability. Besides, the model has a low computation time and memory requirements, which makes it portable for use in IoT sensors and edge computing systems.

5.2 Practical Implications

The practical application of the proposed model is very important for industries that depend on real time decision making. For instance:

1. **Finance:** When it comes to fraud detection, for which the alerts generated in real-time can help to prevent large scale monetary losses, the lightweight model can help pinpoint anomalies in real time correctly.
2. **Healthcare:** To summarize, patient or clients' vital sign data through wearable technology can gain from the proposed effective anomaly detection for immediate notification of an.
3. **Industrial Systems:** In the context of manufacturing, the use of the described model can be applied to identify deviations in machines' performance in order to minimize time spent on maintenance as well as maintenance expenses.
4. **IoT and Smart Cities:** For instance, in smart city context, real-time traffic performance as well as energy consumption can be enhanced through the use of low latency anomaly detection models at the edge.

5.3 Limitations

That being said, there are still drawbacks of the existing anomaly detection techniques; more evidently when it comes to the analysis of the real-time streaming data. Static models, though bear a lot of success in follow traditional paradigms, are a challenge when it comes to be used in stream contexts, where adapt to concept drifts of data distributions are a major concern. Furthermore, there are many machine learning models, which provide high detection accuracy but their implementation involves high computational overhead which becomes a constraint in the application of these algorithms in real-time applications that demand least delay in processing and highest possible throughput. Similarly, the requirement of a large amount of labeled data that is needed in the case of SL techniques limits their practical applicability, as the number of labeled data sets is usually rather limited. Lastly, in efficient, reduced-weight models, high precision could be tricky apart from the fact that such models may not be easily scalable to process large datasets especially for high dimensions.

Despite its strengths, the proposed model has certain limitations:

1. **Limited Scalability for Extremely High-Dimensional Data:** Although the model is advantageous for many real-time systems, it may need further fine-tuning if it is intended to process very high-dimensional datasheets.
2. **Performance in Highly Complex Anomaly Patterns**: The model may sometimes be the inability to discern anomalies which are well camouflaged amidst exceptional temporal structures that may be

present in the data if not fine-tuned or embraced with more sophisticated feature extraction methodologies.

3. **Dependency on Hyper parameters:** The model's outcome depends on proximities of certain hyper parameters in particular, the size of the sliding window for input construction and the threshold for the anomaly scores.

5.4 Future Directions

That being said, there are still drawbacks of the existing anomaly detection techniques; more evidently when it comes to the analysis of the real-time streaming data. Static models, though bear a lot of success in follow traditional paradigms, are a challenge when it comes to be used in stream contexts, where adapt to concept drifts of data distributions are a major concern. Furthermore, there are many machine learning models, which provide high detection accuracy but their implementation involves high computational overhead which becomes a constraint in the application of these algorithms in real-time applications that demand least delay in processing and highest possible throughput. Similarly, the requirement of a large amount of labeled data that is needed in the case of SL techniques limits their practical applicability, as the number of labeled data sets is usually rather limited. Lastly, in efficient, reduced-weight models, high precision could be tricky apart from the fact that such models may not be easily scalable to process large datasets especially for high dimensions.

To address these limitations and further enhance the applicability of the lightweight model, future research could explore:

1. **Hybrid Models:** Enhancing the anomaly detection of highly complex patterns integrating lightweight machine learning into deep learning processes without sacrificing remarkable efficiency.
2. **Auto-Tuning Mechanisms:** Creating live hyper parameters tuning techniques where by hyper parameters are automatically determined according to the type of data stream.
3. **Distributed Processing Frameworks:** Extending the model into distributed streaming frameworks like Apache Kafka or Apache Flink to work for scaling issues involving large streams.
4. **Application-Specific Customization:** Specializing the model for specific industries or kind of data seconds for cybersecurity or measuring environmental impact.

5.5 Broader Implications

The case with the lightweight anomaly detection models (LADs) trend with the modern times of lightweight models in the machine learning (ML) platforms especially in resource constrain environments. Thus, efficiency and scalability of the ML models is more important than ever with the growing importance of real-time results in many industries. The creation of such models will remain highly relevant as we navigate and extend the possibilities of edge computing, IoT, and other IT applications that depend on real-time analytics. These technologies require models that can work in constrained environments, for example, in an embedded microcontroller and remote sensors that possess finite computing and memory capabilities. For this reason, the science of developing and deploying these models will reside as a key role in subsequent development of industries that need time-sensitive decisions and real-time exception handling. With the advancement in edge computing and IoT devices, the next phase of development will focus on

developing better, cost-effective, and accurate models of machine learning to facilitate integration of such technologies into various applications like smart cities, health care and manufacturing and other industries.

Conclusion

Machine learning for real time anomaly detection of streaming time series data has been extremely refreshing especially for industries that need their insights as soon as possible. This work developed a computationally less heavy and efficient ML model for stream data to detect anomalies that work in noisy and perform concept drifts considering the resource limitation of cloud systems.

The findings shown in the study show that the model suggested here has great accuracy, recall, and F1-score indicators with a tiny computational time and space complexity, which satisfies real-time performance demands in finance, healthcare, the IoT, and industrial systems. In contrast to the previously used approaches that fail at the issues of scale and flexibility in handling data streams, the lightweight model was shown to achieve better balance between computational and detection performances.

However, the proposed model is not without its weaknesses: it proves less effective when dealing with very high order of anomaly patterns, and offers less precise control over the tuning of the new set of hyper parameters peculiar to each specific domain. Possible future work: creation of the merged models; incorporation of automatic hyper parameters optimization procedures; unification of maximum model's applicability through the Distributed Computing Frameworks.

Therefore, lightweight machine learning models have significant optimism for use in the real-time anomaly detection of streaming data. When applied in different fields, their use can improve inputs, processes and outcomes by increasing the effectiveness and safeness of systems and organizational environments as components for smarter systems and environments. Thus, this study is among the first to present solutions to streaming data problems that are implementable at a large scale.

Reference

1. Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, *262*, 134-147. https://doi.org/10.1016/j.neucom.2017.04.070
2. Ahmad, S., & Purdy, S. (2016). Real-time anomaly detection for streaming analytics. *arXiv preprint arXiv:1607.02480*. https://doi.org/10.48550/arXiv.1607.02480
3. Habeeb, R. A. A., Nasaruddin, F., Gani, A., Hashem, I. A. T., Ahmed, E., & Imran, M. (2019). Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management*, *45*, 289-307. https://doi.org/10.1016/j.ijinfomgt.2018.08.006
4. Ding, N., Ma, H., Gao, H., Ma, Y., & Tan, G. (2019). Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model. *Computers & Electrical Engineering*, *79*, 106458. https://doi.org/10.1016/j.compeleceng.2019.106458
5. Lee, S., & Kim, H. K. (2019). Adsas: Comprehensive real-time anomaly detection system. In *Information Security Applications: 19th International Conference, WISA 2018, Jeju Island, Korea, August 23–25, 2018, Revised Selected Papers 19* (pp. 29-41). Springer International Publishing. https://doi.org/10.1007/978-3-030-17982-3_3

6. Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003, June). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* (pp. 2-11). https://doi.org/10.1145/882082.882086

7. Lin, J., Keogh, E., & Truppel, W. (2003, June). Clustering of streaming time series is meaningless. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* (pp. 56-65). https://doi.org/10.1145/882082.882096

8. Kavitha, V., & Punithavalli, M. (2010). Clustering time series data stream-a literature survey. *arXiv preprint arXiv:1005.4270*. https://doi.org/10.48550/arXiv.1005.4270

9. Kong, L., & Mamouras, K. (2020). StreamQL: a query language for processing streaming time series. *Proceedings of the ACM on Programming Languages*, *4*(OOPSLA), 1-32. https://doi.org/10.1145/3428251

10. Zhang, M., Guo, J., Li, X., & Jin, R. (2020). Data-driven anomaly detection approach for time-series streaming data. *Sensors*, *20*(19), 5646. https://doi.org/10.3390/s20195646

11. Punithavathi, P., Geetha, S., Karuppiah, M., Islam, S. H., Hassan, M. M., & Choo, K. K. R. (2019). A lightweight machine learning-based authentication framework for smart IoT devices. *Information Sciences*, *484*, 255-268. https://doi.org/10.1016/j.ins.2019.01.073

12. Kumar, A., Arora, H. C., Kapoor, N. R., Mohammed, M. A., Kumar, K., Majumdar, A., & Thinnukool, O. (2022). Compressive strength prediction of lightweight concrete: Machine learning models. *Sustainability*, *14*(4), 2404. https://doi.org/10.3390/su14042404

13. Challapalli, A., Patel, D., & Li, G. (2021). Inverse machine learning framework for optimizing lightweight metamaterials. *Materials & Design*, *208*, 109937. https://doi.org/10.1016/j.matdes.2021.109937

14. Van, N. H., Van Thanh, P., Tran, D. N., & Tran, D. T. (2023). A new model of air quality prediction using lightweight machine learning. *International Journal of Environmental Science and Technology*, *20*(3), 2983-2994. https://doi.org/10.1007/s13762-022-04185-w

15. Hoffpauir, K., Simmons, J., Schmidt, N., Pittala, R., Briggs, I., Makani, S., & Jararweh, Y. (2023). A survey on edge intelligence and lightweight machine learning support for future applications and services. *ACM Journal of Data and Information Quality*, *15*(2), 1-30. https://doi.org/10.1145/3581759

16. Lu, Y., Yu, F., Reddy, M. K. K., & Wang, Y. (2020). Few-shot scene-adaptive anomaly detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16* (pp. 125-141). Springer International Publishing. https://doi.org/10.1007/978-3-319-71249-9_3

17. Vávra, J., Hromada, M., Lukáš, L., & Dworzecki, J. (2021). Adaptive anomaly detection system based on machine learning algorithms in an industrial control environment. *International Journal of Critical Infrastructure Protection*, *34*, 100446. https://doi.org/10.1016/j.ijcip.2021.100446

18. Yahaya, S. W., Lotfi, A., & Mahmud, M. (2021). Towards a data-driven adaptive anomaly detection system for human activity. *Pattern Recognition Letters*, *145*, 200-207. https://doi.org/10.1016/j.patrec.2021.02.006

19. Alkahtani, H., Aldhyani, T. H., & Al-Yaari, M. (2020). [Retracted] Adaptive Anomaly Detection Framework Model Objects in Cyberspace. *Applied Bionics and Biomechanics*, *2020*(1), 6660489. https://doi.org/10.1155/2020/6660489

20. Dong, L. I., Shulin, L. I. U., & Zhang, H. (2017). A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples. *Pattern Recognition*, *64*, 374-385. https://doi.org/10.1016/j.patcog.2016.11.026