ISSN: 2756-2638
Transactions on Engineering and Computing Sciences; Vol. 14 No. 06 (2026) (01-08)
https://doi.org/10.14738/tecs.1406.12653

A Brief Study: Anomaly Detection of Network Traffic Using Isolation Forest in a Big Data Platform Using the Kafka Pipeline

Sikha S. Bagui[1]*, Ben Yanick[1], Dustin Mink[2], and Subhash C. Bagui[3]

1. Department of Computer Science, The University of West Florida, Pensacola, Florida, USA
2. Department of Cybersecurity, The University of West Florida, Pensacola, Florida, USA
3. Department of Mathematics and Statistics, The University of West Florida, Pensacola, Florida, USA

Abstract: In today's digital world, data is continuously being produced at an extremely fast rate. Data that is generated in a continuous process is referred to as streaming data. The challenge in working with streaming data is that it requires an environment that can process and store massive amounts of data efficiently. This work addresses the challenges of using streaming data using Kakfa, a distributed streaming platform allowing real-time data processing in the Big Data environment. Of course, streaming data can also contain anomalies. Hence, in this work, a machine learning algorithm, Isolation Forest, is integrated with Kafka to detect anomalies in streaming data.

Keywords: Anomaly Detection, Streaming Data, Zeek Data, UWF-ZeekData22, Kafka, Big Data

INTRODUCTION

In today's digital world, data is produced at an extremely fast rate. Everything from social media interactions to sensor readings in smart devices to raw network logs generates data in real time, continuously generating more and more data. Data that is generated in a continuous process is referred to as streaming data. Streaming data is characterized by its velocity, volume, and variety, requiring real-time processing and analysis to extract valuable insights and detect anomalies as they occur. With streaming data constantly being produced, we need to determine whether all this data is good data or if it contains cybersecurity threats.

The challenge in working with streaming data is that it requires an environment that can process and store massive amounts of data efficiently, hence the need for a Big Data environment. The volume of data available in streaming data is far greater than the available memory. Apache Kafka [1] is a streaming data store in the Big Data environment that can efficiently process massive amounts of data while also allowing machine learning to be incorporated into the data analysis process [2].

Within streaming data, there can be anomalies. Anomalies are data that deviate significantly from the norm. Anomalies could be as simple as the system needing preventive maintenance or as complicated as being intentionally fraudulent data and cyber threats. The objective of this paper is to handle intentional cyber threats in streaming data. To tackle these problems, we need effective machine learning tools.

In this work, to detect anomalies, a machine learning algorithm, Isolation Forest, is integrated with Kafka. Isolation Forest operates on the principle of isolating anomalies instead of profiling normal data, making it particularly powerful for real-time streaming applications [3].

Streaming platforms such as Apache Kafka [1] have revolutionized real-time data processing and analysis. Kafka's ability to handle high-throughput data streams and its architecture provide a robust infrastructure for implementing machine learning models that require continuous updates [2]. While some anomaly detection frameworks have been reviewed in Kafka environments, the concrete integration of Isolation Forest for dynamic anomaly detection has not yet been done. This study fills this gap by using Isolation Forest with Kafka to provide a seamless, adaptive approach to anomaly detection in streaming data. This approach not only improves the machine learning algorithm's detection capabilities by updating it as new data is ingested, but also ensures that the model remains relevant and accurate over time. Our research contributes to this knowledge by demonstrating how effective the integration of an Isolation Forest model can be as a semi-supervised model with Kafka, paving the way for future studies in real-time anomaly detection with streaming data.

The rest of this paper is organized as follows. Section 2 presents the background of Kafka and Isolation Forest; section 3 presents the related works; section 4 presents the data and methodology; section 5 presents the results; section 6 presents the discussion; section 7 presents the conclusions, and finally section 8 presents the future works.

BACKGROUND

This section briefly introduces Kafka, Kafka Topics, the Kafka Publisher-Subscriber Model and Isolation Forest.

Kafka

The streaming aspect of the data used in this research was handled using Apache Kafka [1], a distributed streaming platform allowing real-time data processing. Being a distributed streaming platform, Apache Kafka can handle large amounts of data without latency issues. Taking on the role of a message broker, other systems, like Isolation Forest, can subscribe to streams of records in a publisher-subscriber manner. Since multiple subscribers can be supported in Kafka, scalability of the pipeline is possible [2].

Kafka Topics

A Kafka topic serves as a category or feed to which Kafka producers send records (or messages) and from which Kafka consumers read. Topics are divided into one or more partitions, each functioning like an append-only log where new records are added sequentially. Each record within a partition has a unique identifier called an offset, which helps consumers keep track of the messages they have processed. Consumers can start reading from any offset, allowing them to reprocess data if needed.

Kafka Publisher-Subscriber Model

With many message brokers such as Kafka, a common approach to connecting data produced with data consuming devices/applications is the publisher-subscriber (pub-sub) mode. This design marks data sources as publishers and stores the data in a structure such as a Kafka topic [4]. Kafka consumers can subscribe to multiple topics or multiple consumers can subscribe to a single topic [4]. This makes message passing between devices effective when handling streaming data.

Isolation Forest

To evaluate network traffic, the machine learning algorithm, Isolation Forest [5], was used to monitor traffic for threat actors (anomalies) on the network. The Isolation Forest model uses an unsupervised machine learning approach for anomaly detection. This is achieved through isolating observations in a dataset by randomly selecting features and splitting data points based on those features. The model holds one assumption, that is, anomalies are few and different. Therefore, isolation of anomalies occurs early in the process of splitting data points. How the algorithm handles this splitting of data is by constructing a series of decision trees, where data points that are easier to isolate (i.e. anomalies) require fewer splits (fewer levels in tree). This approach makes Isolation Forest efficient and scalable, especially for detecting outliers in large datasets.

RELATED WORKS

The growth of streaming data has fostered considerable interest in the development of robust anomaly detection techniques, especially in the areas of real-time analytics and machine learning. In [6] Kareem and Muhammed (2024) focused on enhancing the anomaly detection process in data streams by exploring the effectiveness of modifications to the Isolation Forest algorithm. They looked into how changing the structure or building methods of isolation trees could improve the ways we find odd things in complex data streams. The research also presented recent improvements. Some of these used clustering or autoencoders with isolation trees.

In [7] Xu et al. (2017) focused on improving the Isolation Forest algorithm for anomaly detection by addressing its limitations, such as redundancy in iTrees, low accuracy, and high memory usage. The SA-iForest method combines simulated annealing with iTree selection to improve anomaly detection performance. Using this approach improves accuracy, efficiency, and reliability, especially when working with massive amounts of data.

In [8], Melquiades and Keto (2022) showed a new semi-supervised anomaly detection model called the Multi-Class Hybrid Isolation Forest (MCHIF). It builds on the traditional Isolation Forest's ability to detect outliers through isolation-based scoring. Unlike the standard Isolation Forest and Hybrid Isolation Forest (HIF) methods, which can only identify a single class of anomalies, MCHIF uses labeled data to detect many types of anomalies. This makes the model better at distinguishing between different anomaly types, all while maintaining the speed and scalability of the original algorithm.

Moomtaheen et al. (2024) [9] extended the Isolation Forest algorithm to deal with Big Data in Cybersecurity Intrusion Detection, but they did not deal with streaming data in using Kafka.

THE DATA AND METHODOLOGY

Zeek Data

The Zeek data used in this work, UWF-ZeekData22 [10,11], created at the University of West Florida (UWF) for cybersecurity research, consists of network traffic logs labeled according to the MITRE ATT&CK Framework [12]. Zeek is an open-source network analysis framework that is used for monitoring network traffic and conducting security analysis. Zeek analyzes network traffic in real-time and generates logs for various network protocols (i.e. HTTP/HTTPS, UDP, etc.) The dataset is composed of approximately over 18 million records with 23 fields [10].

Data Pipeline - Kafka for Real-Time Streaming Data

Apache Kafka was used to handle the data pipeline for real-time streaming data by passing the Zeek log data to a downstream analysis. The logs were transformed into Kafka topics. Kafka producers were responsible for sending these Kafka topics to a Kakfa consumer to analyze the data. Overall, Kafka allows data to be brought into the pipeline and analyzed without significant delays, which is important for real-time anomaly detection.

Anomaly Detection with Isolation Forest

The Isolation Forest algorithm is a tree-based model that isolates and identifies anomalies in large datasets [5]. The algorithm creates multiple binary decision trees by recursively splitting the data along randomly chosen features and split values. It provides a predictive score and identifies a datapoint as an anomaly.

As for anomaly detection in the streaming process: First, for data preprocessing, the following features were selected: “Mitre_Attack_Tactics”, “orig_ip_bytes”, “duration”, “orig_pkts”, “resp_ip_bytes”, and “dest_port.”. Second, the model was trained with 70% of the data. Third, the Isolation Forest model assigned an anomaly score, which indicated how likely the event was to be anomalous. Events with high anomaly scores were flagged for further investigation.

Real-Time Anomaly Detection Workflow

The real-time anomaly detection workflow consists of a Kafka Producer, Kafka Consumer, Isolation Forest Model, and an Alerting System. The producer pushes Zeek log files into a Kafka topic, which is later consumed by a Kafka Consumer [2]. The producer is also responsible for ensuring these topics are sent to Kafka consumers with as little delay as possible. A consumer reads a Kafka topic sent by the producer and processes the data in real time [2]. The consumer converts raw log files into feature vectors, allowing the Isolation Forest model to analyze them. The Isolation Forest model identifies anomalies based on the

trained decision trees [5]. Lastly, the alerting system is a downstream system that can be alerted if the Isolation Forest model detects an anomaly, allowing investigation of potential security events.

Evaluation and Performance Metrics

Precision and recall are used to assess the effectiveness of detecting true anomalies while minimizing false positives. The F1-score provides a balanced measure of both precision and recall.

RESULTS

Table 1 presents the performance metrics for different anomaly threshold values. It can be noted that the best results were at threshold value of 0.411. Table 2 presents the True Positives, True Negatives, False Positives and False Negatives. From Table 2, too, it can be observed that the lowest number of false positives and false negatives were at the same threshold of 0.411.

Table 1: Performance metrics for different anomaly threshold values

Threshold	Accuracy	Precision	Recall	F1-Score	AUC
0.425	0.847	0.776	0.875	0.65	0.833
0.42	0.85	0.786	0.885	0.805	0.833
0.411	0.853	0.795	0.91	0.823	0.833
0.405	0.85	0.78	0.905	0.815	0.833
0.4	0.847	0.788	0.902	0.808	0.833

Table 2: True/false positive and negative counts for different anomaly threshold values

Threshold	True Positives	True Negatives	False Positives	False Negatives
0.425	1287	4371	936	106
0.42	1316	4365	943	77
0.411	1394	4344	963	0
0.405	1394	4309	998	0
0.4	1394	4235	1072	0

Figure 2 presents the anomaly scores. Table 3 presents the counts of samples at different anomaly ranges. Table 4 presents a statistical analysis of anomaly scores for the training data.

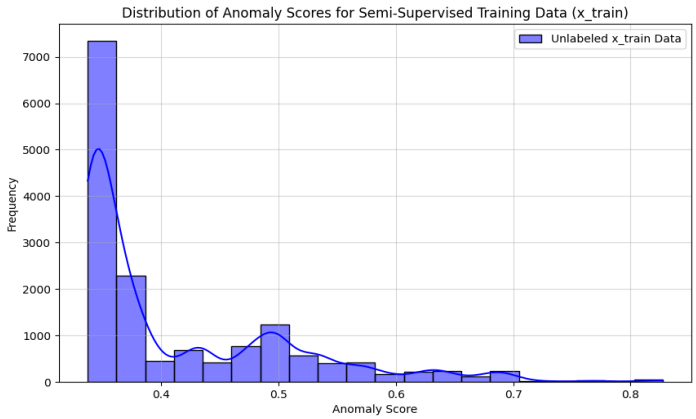


Figure 2: Plot of the predicted scores of the training data.

Table 3: Counts of samples within specified anomaly score range.

Data Points	Count
0.3-0.5	12647
0.5-0.7	2876
<0.3 or > 0.7	113

Table 4: Statistical Analysis of anomaly scores for the training data.

Statistic	Output
Mean	0.413
Min	0.337
Max	0.827
Median	0.365

DISCUSSION

The results in Tables 1-4 give a comprehensive performance evaluation of the Isolation Forest model for anomaly detection, proving its robustness as a semi-supervised approach. First, threshold optimization is considered by investigating the performance metrics accuracy, precision, recall, F1-score, and AUC over the threshold range from 0.400 to 0.425. A clear trend appears whereby, as the threshold decreases, recall remains consistently high ≥ 0.875, whereas precision slightly decreases. This indicates that the model prefers to increase recall, meaning more true anomalies are found, though at the cost of a few more false positives. The AUC for all thresholds has constantly been at 0.833, reflecting that this model is reliable in distinguishing between normal data and anomalous. The F1-score reaches its peak at 0.411 with a score of 0.823, the best trade-off between precision and recall for applications using this approach.

Analysis of the confusion matrix also shows the effects of adjusting thresholds. Gradually lowering thresholds increase true positives, which means better anomaly detection. The tradeoff is a higher false positive rate due to the lenient decision boundary. Importantly, the false negatives drastically decrease from 106 for a threshold of 0.425 to zero when thresholds of 0.411 or lower are reached. This ensures that anomalies are not

missed. This reduction puts in evidence that careful threshold selection is important, especially in tasks where the missing anomalies have a serious impact. This reduction showcases the importance of a very carefully chosen threshold, especially in real-world applications where missed anomalies are extremely costly. Such trade-offs are typical of semi-supervised anomaly detection models, where identifying anomalies is prioritized, even at the expense of increased false alarms.

The distribution of the predictive scores, as explained in Table 3, is such that about 81% of the data points fall within the range of 0.3 to 0.5. Only a small fraction (0.7%) falls at the extreme ends (<0.3 or >0.7), representing outliers or highly confident predictions. This reflects that most samples have a moderate likelihood of being anomalies, which is as expected with Isolation Forest, which uses depth-based scoring to segregate anomalies from normal data. From a statistical point of view, the mean predictive score of 0.413, with a median value of 0.365, indicates a slight skew to higher scores, as seen in Table 4. The maximum score of 0.827 and the narrow range suggest that the model avoids over estimation of anomaly probabilities, instead giving delicate separations between normal and anomalous data.

CONCLUSION

In summary, this model proposal puts in a very good performance with the Isolation Forest. It achieved an AUC of 0.833 for all cases, with an optimum threshold of 0.411 that gave a balanced precision and recall. With an F1-score high of 0.823 at no anomaly missed, it is reliable even under critical conditions. When combined with Apache Kafka, the Isolation Forest's ability to detect anomalies in streaming data improved significantly. Kafka's high-throughput and fault-tolerant design ensures smooth data delivery for continuous evaluation, enabling real-time insights. This setup is well-suited for managing big data, supporting ongoing learning, and boosting operational efficiency, making it a robust solution for fast-paced, data-rich environments.

FUTURE WORK

With the use of streaming data in systems becoming more and more prevalent these days, the continuous efficacy and agility of anomaly detection models such as Isolation Forest become more important. Further improvements could come with integrating other machine learning techniques to further improve scalability of Kafka stream processing with the Isolation Forrest model and fine-tuning the model for specific use cases. This combination positions the Isolation Forest model as a powerful tool for modern anomaly detection challenges.

REFERENCES

[1]. https://kafka.apache.org/

[2]. Carnero, C. Martín, G. Jeon, and M. Díaz, "Online learning and continuous model upgrading with data streams through the Kafka-ML framework," Future Generation Computer Systems, vol. 160, pp. 251–263, 2024.

[3]. R. Sharma and M. Grover, “Enhancing Cybersecurity with Machine Learning: Evaluating the Efficacy of Isolation Forests and Autoencoders in Anomaly Detection,” in 2024 International Conference on Circuit Power and Computing Technologies (ICCPCT), pp. 1017–1021, 2024.

[4]. Amazon Web Services, “What is Pub/Sub Messaging?,” [Online]. Available: https://aws.amazon.com/what-is/pub-sub-messaging/. [Accessed: Aug. 22, 2024].

[5]. F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based Anomaly Detection,” in ACM Transactions on Knowledge Discovery from Data, vol. V, no. N.

[6]. M. S. Kareem and L. A. Muhammed, "Anomaly Detection in Streaming Data using Isolation Forest," in 2024 Seventh International Women in Data Science Conference at Prince Sultan University (WiDS PSU), pp. 223–228, 2024.

[7]. D. Xu, Y. Wang, Y. Meng, and Z. Zhang, "An Improved Data Anomaly Detection Method Based on Isolation Forest," in 2017 10th International Symposium on Computational Intelligence and Design, pp. 287–291, 2017.

[8]. Melquiades and F. B. de Lima Neto, “Isolation Forest-based Semi-supervised Anomaly Detection of Multiple Classes,” in 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, 2022.

[9]. Moomtaheen, F., Bagui, S.S., Bagui, S.C. and Mink, D. (2024). Extended Isolation Forest for Intrusion Detection in Zeek Data. Information, 15, 404. https://doi.org/10.3390/info15070404

[10]. Bagui, S.S., Mink, D., Bagui, S.C., Ghosh, T., Plenkers, R., McElroy, T., Dulaney, S. & Shabanali, S. (2023). Introducing UWF-ZeekData22: A Comprehensive Network Traffic Dataset Based on the MITRE ATT&CK Framework. Data, 2023, 8, 18. https://doi.org/10.3390/ data8010018.

[11]. Datasets.uwf.edu.edu

[12]. https://attack.mitre.org/

ACKNOWLEDGEMENTS

This work has been partially supported by the Askew Institute of the University of West Florida.