IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 June 2024, accepted 28 June 2024, date of publication 9 July 2024, date of current version 19 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3425472

RESEARCH ARTICLE

SXAD: Shapely eXplainable AI-Based Anomaly Detection Using Log Data

KASHIF ALAM[1], KASHIF KIFAYAT[2], GABRIEL AVELINO SAMPEDRO[3], (Member, IEEE), VINCENT KAROVIČ JR.[4], AND TARIQ NAEEM[1]

[1]Department of Computer Science, Faculty of Computing and AI, Air University, Islamabad 44000, Pakistan
[2]Department of Cyber Security, Faculty of Computing and AI, Air University, Islamabad 44000, Pakistan
[3]School of Management and Information Technology, De La Salle-College of Saint Benilde, Manila 1004, Philippines
[4]Department of Information Management and Business Systems, Faculty of Management, Comenius University Bratislava, 82005 Bratislava, Slovakia

Corresponding authors: Kashif Alam (kashifalam1@gmail.com) and Vincent Karovič Jr. (vincent.karovic6@fm.uniba.sk)

ABSTRACT Artificial Intelligence (AI) has made tremendous progress in anomaly detection. However, AI models work as a black-box, making it challenging to provide reasoning behind their judgments in a Log Anomaly Detection (LAD). To the rescue, Explainable Artificial Intelligence (XAI) improves system log analysis. It follows a white-box model for transparency, understandability, trustworthiness, and dependability of Machine Learning (ML) and Deep Learning (DL) Models. In addition, Shapely Additive Explanation (SHAP), added to system dynamics, makes informed judgments and adoptable proactive methods to optimize system functionality and reliability. Therefore, this paper proposed the Shapely eXplainable Anomaly Detection (SXAD) framework to identify different events (features) that impact the models' interpretability, trustworthiness, and explainability. The framework utilizes the Kernel SHAP approach, which is based on Shapley values principle, providing an innovative approach to event selection and identifying specific events causing abnormal behavior. This study addresses the LAD by transforming it from a black-box model into a white-box one, leveraging XAI to make it transparent, interpretable, explainable, and dependable. It utilizes benchmark data from the Hadoop Distributed File System (HDFS), organized using a Drain parser, and employs several ML models, such as Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB). These models achieve impressive accuracy rates of 99.99%, 99.85%, and 99.99%, respectively. Our contribution are novel because no earlier work has been done in the area of Log Anomaly Detection (LAD) with integration of XAI-SHAP.

INDEX TERMS Explainable artificial intelligence, Shapley additive explanation, Hadoop distributed file system, machine learning.

I. INTRODUCTION

Anomaly is a focal point in Machine Learning (ML) and data mining [1]. Its significance extends across diverse sectors, including cybersecurity, financial resources, manufacturing, and energy [2]. It plays a pivotal role in AI applications, such as network security, fraud detection, healthcare, energy, event prediction, program verification, and problem diagnosis [3]. In such applications, log analysis is essential in detecting malicious activities, investigating security incidents, and identifying potential vulnerabilities in systems and networks [4], [5], [6]. Analyzing these logs, anomaly detection identifies deviations from normal behavior in a data set. However, most of the ML-based anomaly detection techniques work as a black-box [7]. Therefore, a novel field, XAI [8] addresses this issue by enhancing the transparency, accountability, interpretability, and trustworthiness of ML models. DARPA launched the ''Explainable AI (XAI) Program'' in early 2017 to develop ML models that are both highperforming and comprehensible to human users, enabling them to trust and manage AI systems and similarly, NIST introduced four ground principles and rules for XAI systems in 2020 [9]. These principles include the explanation, meaningful, accuracy, and knowledge limitation.

The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Ayyash.

Different methods and techniques have been developed to make AI models explainable. They work with a distinction between interpretability, explainability, transparency, and trustworthy models, which are distinguished from post-hoc interpretations These are additional methods used to ensure transparency for multifaceted black-box models. These models integrate both local and global justification generation for individual input or for the entire model [10]. Industry 5.0 principles require AI models to be interpretable, human-in-the-loop, and transparent [11]. Anomaly detection and system failure categorization are critical components of predictive maintenance [12]. It enables organizations to reduce system downtime, improvement of maintenance calendars and enhances operational performances. Maintenance professionals may avoid unexpected failure by applying machine learning approaches to predict when a system will fail. Several problems arise as a result of the absence of explanation in Log Anomaly Detection (LAD) [13], [14]. First, it is important to note that end users may need help comprehending the detection models. Second, there is sometimes a disconnect between forecasts and explanations because ML models frequently make predictions without providing explicit explanations for the judgments they make. Third, there needs to be more transparency in the prediction process, making it challenging to understand the reasoning behind the predictions. Finally, issues of unfairness need addressability in the prediction process [3], [15]. In this regard, XAI aims to develop AI models with trust and accuracy [16]. It uses both extrinsic and intrinsic techniques to validate the model's forecasts. It also entails creating algorithms and procedures for creating understandable and transparent AI models.

Similarly, Explainable Anomaly Detection (XAD) is the process of gaining valuable insights from a model designed to detect anomalies. These insights are pertinent to the correlations identified within the data or obtained via the model. The significance of this information lies in its capacity to provide significant insights into the anomaly detection issues that the end user is investigating [17]. An overview of XAD and traditional ML models are shown in Figure 1. Therefore, this study aims to move from black-box models to white-box models, making the LAD process more transparent, interpretable, explainable, and reliable making it SXAD. Various ML models are implemented using benchmark system logs data HDFS as proof of the effectiveness of SXAD. However, to the best of our knowledge, efforts have not been made to find the main causes of anomalous events forecasted by the model in LAD for transparency, interpretability, and explainability. Therefore, this research aims to fill this gap by conducting a comprehensive analysis of the anomalous events detected by the model in LAD, with a focus on identifying the underlying causes. By uncovering the factors contributing to anomalous predictions, we seek to enhance the transparency, interpretability, and explainability of the model's outputs, thus providing valuable insights for system administrators and stakeholders.

The contributions of our study are presented below:

- The proposed framework enhances the performance of ML black box model in identifying and forecasting large-scale system failure using SHAP which improve understandability, transparency and trustworthiness in LAD.
- Using the Kernel Explainer approach, we effectively identified key events as feature in log data towards prediction of system failure.
- We evaluate the impact and performance of our methodology by compared it with existing methods thereby offering a comprehensive understanding of its overall effectiveness.
- Our contribution demonstrate the importance of XAI in identifying system log anomalies as a essential step towards achieving Industry 5.0 goals for large scale system failure prevention.

The remaining paper is divided into different sections. Section II analyzes the literature review. Section III describes the suggested framework. Section IV discusses the results, evaluation, and performance metrics. Section V concludes the study and VI provides the future directions.

II. RELATED WORKS

Anomaly detection analyses log data of heterogeneous systems such as system logs [4], [18], [19], [20], [21], event logs [22] and various other logs [23], [24], [25], [26], [27]. The anomaly detection techniques find anomalous behaviors in logs that depart from conventional systems. There are two major classes with approaches to logs: supervised and unsupervised models. The former requires training data collection with labeled examples for regular and anomaly classes. It is not easy to access reliable representative models for anomaly classes. A different kind is the no-training-data-required unsupervised LAD model. In [18], the researchers explored the link between sequences of system logs and behavior patterns to detect anomalies within Hadoop data. This approach improved F-Score by 13%. The study needed more explanations and transparency concerning the results obtained. A similar study, [19], utilized ML with the Word2Vec algorithm for log mining. The technical approach included RF, MLP, and Gaussian NB. The achieved accuracy was 90%. The study need to offer explanations and transparency regarding the results that were obtained.

Similarly, [20] focused on anomaly detection in big data system logs using DL. The technical approach involved Log Parsing and logkey2vec techniques, with Convolutional Neural Network (CNN) employed. The achieved performance metrics included precision, recall, and F1-measure of 95%, 95%, and 96%, respectively. The study should have explained the transparency regarding the obtained results. Wang et al. [21] applied a DL-based approach for anomaly detection in system logs. They technically included the use of TF-IDF for preprocessing and feature extraction, with Long Short-Term Memory (LSTM) used as a model. The achieved performance metrics included an impressive F1-score of 0.99. However,

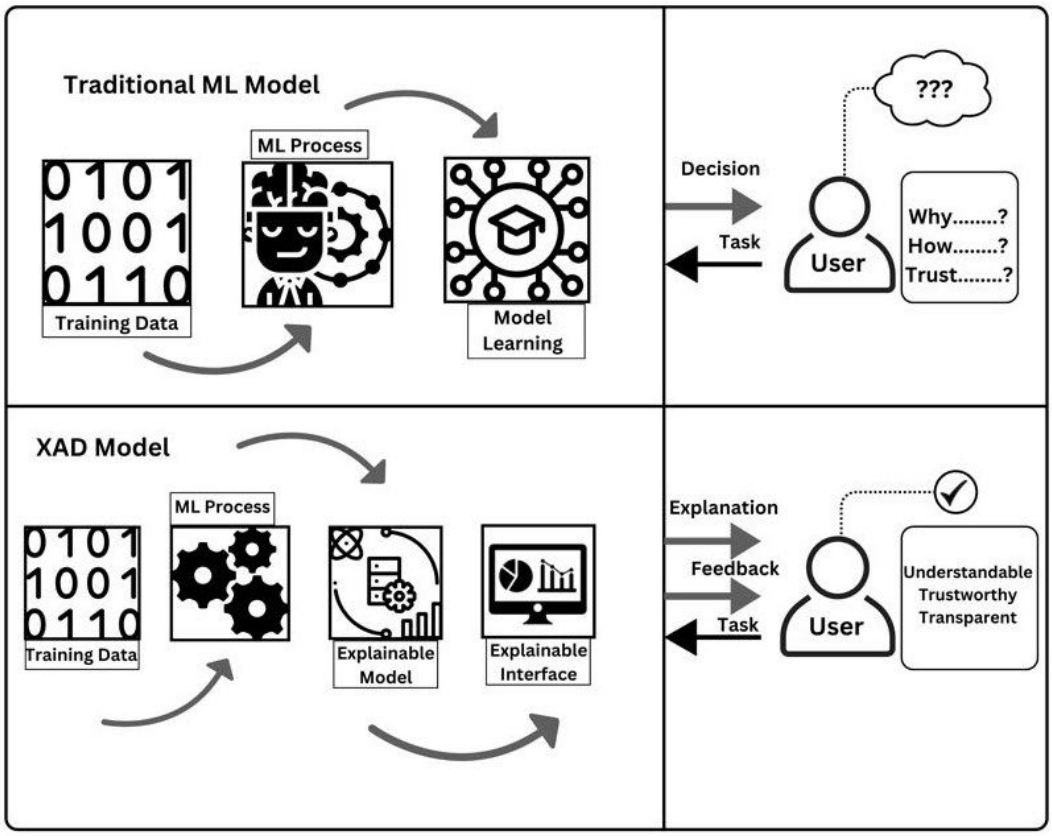

FIGURE 1. An Overview of XAD and traditional ML model.

the study did not provide explanations as well as transparency regarding the results. On the other hand, the study needed to provide more interpretability and transparency about is findings.

Log anomaly is an unsupervised technique that was designed to discover anomalies in unstructured logs sequentially and quantitatively. In [22], the authors examined log anomaly using LSTM. An approach for classifying log anomalies, which is independent of the device being utilized, with a specific focus on switch log data [23]. ML techniques were incorporated TF-IDF for preprocessing and positive-unlabeled for learning Support Vector Machine (SVM) for modeling. Notably, this approach showcased remarkable performance, achieving a notable F1 score of 99.51%, along with Macro-F1 of 95.32% and Micro-F1 of 99.74%. However, the study did not provide explanations in relation to its findings.

In [24], a ''Crude'' method is proposed to enhance the detection of error accuracy in distributed systems on a big scale by combining console logs and usage resource data. The approach involved clustering, employing mutual information and entropy for preprocessing, and implementing Hierarchical clustering with PCA. The achieved true positive rate was 80%. However, the study did not provide explanations, interpretability, or transparency regarding its findings. In addition, Jia et al. [25], proposed anomaly diagnosis through a method called ''Logsed,'' which involves mining control graphs that are time-weighted in transactional operational logs. The technical approach encompassed control flow graph and template mining, while the approach to feature extraction involved time-weighted control flow graphs. The method achieved a Precision and Recall of 80%. However, the study did not provide explanations, interpretability, or transparency regarding its findings. In [26], LogEvent2vec, a method for identifying anomalies in extensive IoT logs, is proposed. For parsing and preprocessing, ML techniques such as Drain and Word2Vec are employed. In terms of modeling, Neural Networks (NN), RF, and Naive Bayes (NB) are utilized. The approach yielded remarkable outcomes, with a precision of 94.7%, a recall of 94.0%, and an F1 measure of 0.94. Nevertheless, the study needed more elucidation, comprehensibility, and clarity about its results. Table 1 presents quick access to the state-of-the-art comparison of various techniques.

In [27], a cluster-based approach is presented. It organizes feature vectors of logs into groups with high similarity across members of the same cluster as opposed to vectors from different clusters. Atypical clusters comprise only a handful of data points. Xie et al. [28] the focus is on developing a guided model with confidence for addressing anti-concept

TABLE 1. Comparison among state-of-the-art techniques.

Ref.	Dataset	Preprocessing	Techniques	Performance	Explanation	Transparency
[4]	System Log	Log Parsing	LSTM	TP:100%, FP:38.2% to 1.1% for 10% data	No	No
[18]	System Log	Erroneous Behavior	Clustering	F-Score Improved by 13%	No	No
[19]	System Log	Word2Vec Algorithm	RF, MLP	Accuracy:90%	No	No
[20]	System Log	Log Parsing logkey2vec	CNN	Precision 95% Recall 95, F1-Measure 96	No	No
[21]	System Log	Word2vec, TF-IDF	LSTM	Accuracy, Precision, Recall, F1-score: 0.99	No	No
[22]	System Log	Template2Vec	LSTM	Precision 0.95 Recall 0.94 F1 Score:94	No	No
[23]	Switch Log	TF-IDF	SVM	F1 score: 99.51, Macro-F1: 95.32 Micro-F1: 99.74	No	No
[24]	Console Log	MutualInformation Entropy	PCA	True Positive rate 80%	No	No
[25]	Transactional Log	Template Mining	Graph	Precision, Recall: 80%	No	No
[26]	Event Log	Drain	RF, NB, NN	Precision: 94.7% Recall: 94.0% F1 Measure: 0.94	No	No
[27]	System Log	Drain	LR,DT,SVM	Precision, Recall, F1-score: 0.99%	No	No
[28]	System Log	Parsing	Confidence Guided Parameter	Precision: 98.2% Recall: 95.2% F1 Measure: 96.7%	No	No
[29]	Execution Log	Template Mining	Graph	Average Precision: 90% Recall =80%	No	No
Proposed Method	System Log	TF-IDF and Mean	DT, RF, GB	Precision,Recall,F1 Measure = 0.99%	Yes	Yes

drift in dynamic logs. A precision of 98.2%, a recall of 95.2%, and an F1-measure as high as 96.7% were among the performance criteria that were accomplished.

In [30] proposed LogCluster, a system for organizing log sequences that handle atypical sequences to help developers spot issues at a glance. The cluster centroid determines the representative sequence. Furthermore, in [31], the authors proposed the Log3C framework to combine system Key Performance indicators (KPIs) to identify significant issues in service systems. For efficiency, they suggested a cascading clustering approach. At last, they employ a multivariate linear regression model to pinpoint the critical factors contributing to a decline in KPIs.

Several DL based methods for detecting log anomalies have appeared in recent years. In [32], the authors proposed the LogRobust for information retrieval from unstable log. In [33], the authors utilized Natural Language Processing (NLP) and Information Retrieval (IR) methods to get informative data from logs. In [34], the author used a Generative Adversarial Network (LogGAN) for construct a lack of abnormal data. Similarly, other AI models have also been used in the literature, such as transfer learning [35], and federated learning (FL) [36]. In order to investigate the "black box" nature of ML and the expansion of AI across a range of fields, XAI has been implemented in a number of areas, including cyber security, finance, health care, and industry [8]. Explainable Artificial Intelligence (XAI) increases the interpretability and transparency of anomaly detection models [7]. However, to the best of our knowledge, efforts have not yet been made to find the main cause of anomalous events forecasted by the models in LAD for transparency, interpretability, and explainability. Therefore, this research aims to fill this gap.

III. PROPOSE METHODOLOGY

This study proposes a framework called Shapely eXplainable Anomaly Detection (SXAD) for log data analysis. Though the selection of DT, RF, and GB in our proposed methodology has their unique strengths, DT offers transparency in decision-making, while RF tackles high dimensionality and overfitting, and GB excels in predictive accuracy [37], [38]. We use multi-model approach that ensures a robust and versatile model, backed by empirical evidence of success in HDFS dataset. By leveraging the strengths of these techniques, we aim to deliver both accuracy and interpretability, tailored to our research objectives. RF and GB can become very complex models, particularly when used in ensemble fashion, potentially leading to overfitting. Using a simpler DT alongside them can help balance model complexity with performance. Each individual DT in the ensemble can be analyzed to understand how it makes decisions and which features it considers most important. Figure 2 illustrates the proposed SXAD framework, which consists of four phases: (1) data preparation, (2 models used, (3) log anomaly detection, and (4) explanation using XAI-SHAP. A step-wise methodology is mentioned below:

A. LOGS

Logs provide the essential and required data for an anomaly detection mechanism. The steps involved in the anomaly detection of logs include:

1) Collection: The large data-intensive systems frequently produce logs,these logs typically include a timestamp and a log message describing what happen at that moment. Logs are collected first for subsequent use since they include useful information. Logs play a crucial role in various fields, including anomaly detection, software development, system administration, and more [19], [20], [21], [22].They give significant insights into the working of systems, applications, and networks.
2) Dataset: Real-world log records, especially ones with tags used for evaluation, are either limited or mostly kept secret. Loghub [39] is a free place to store

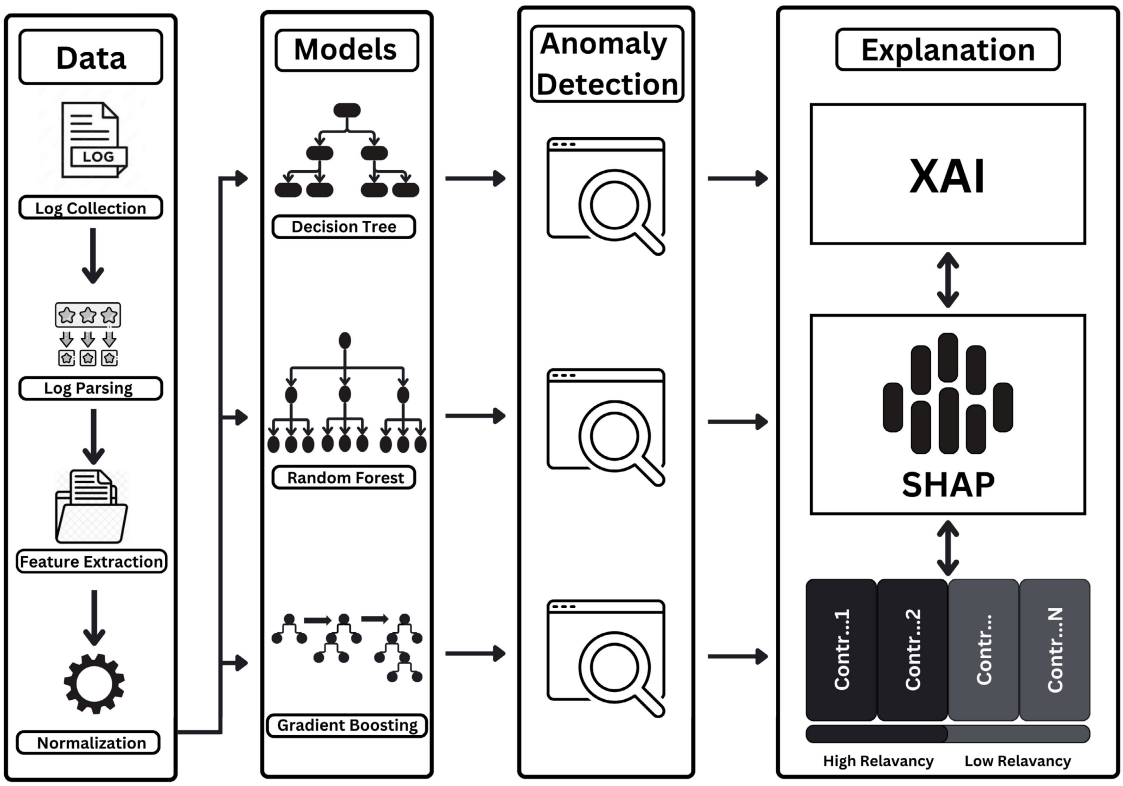

FIGURE 2. The proposed SXAD framework.

and study sixteen (16) different types of logs from distributed systems, super-computers, OS, cellular devices, servers, and stand-alone software. We use the HDFS log because it is the most researched dataset labeled by various researchers. HDFS consists of logs made by Hadoop. These instances run on over 200 Amazon EC2 nodes spread across the network. It has a total of 575,0561 blocks in the log dataset,among which 16,838 were recorded as anomalous by Hadoop experts [40], [41]. HDFS is a distributed file system specifically designed to handle large-scale data processing tasks on commodity hardware. Its ability to scale across hundreds, or even thousands, of nodes makes it a cornerstone technology for organizations grappling with massive volumes of data. By utilizing the HDFS dataset in our research, we directly engage with a crucial aspect of big data management, showcasing the applicability of our findings in real-world scenarios. Furthermore, our utilization of DT, RF, and GB algorithms on the HDFS dataset underscores the practical significance of our approach. These algorithms are widely employed in various industries to extract actionable insights from complex datasets. By demonstrating the effectiveness of these algorithms on the HDFS dataset, we not only validate the relevance of our research but also provide valuable insights that can inform decision-making processes in data-intensive environments [42]. Our research endeavors to bridge the gap between theoretical frameworks and practical applications by employing cutting-edge algorithms on the HDFS dataset. By showcasing the utility of our approach in addressing real-world challenges, we aim to contribute meaningfully to both academic discourse and industry practices.

3) Log Parsing: Data in unstructured logs is typically free-form. Log parsing and obtaining a library of event templates are necessary to facilitate the classification of unstructured logs. Depending on a user-specified set of parameters, each log entry can be converted into a specific event template (constant component) (variable part). Figure 3 shows log parsing action. It illustrates the block size with block ID and event numbers processed by log parser from unstructured logs. We used Drain [43], an efficient log parser, to log parsing in the proposed framework.

B. FEATURE EXTRACTION

We encode the events in the parsed logs to employ ML models as numerical feature vectors our data approach uses a well-known open-source toolkit [27]. We apply the Term Frequency Inverse Document Frequency (TF-IDF). It is a numerical statistic used in NLP and IR techniques for feature extraction [44].

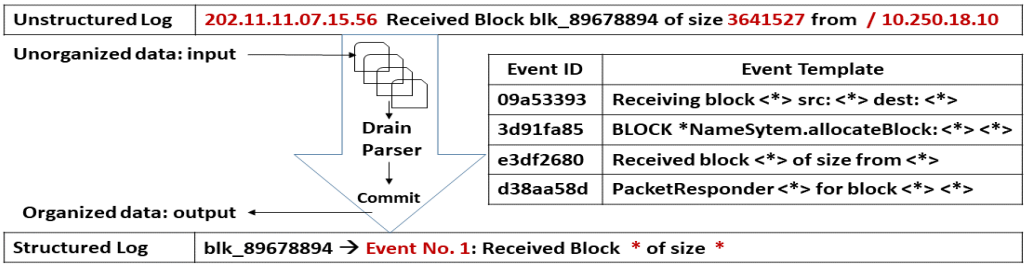

FIGURE 3. An insight of log parsing.

C. NORMALIZATION

Normalization techniques are used to measure the significance of a word within a document or a corpora by considering how frequently the term appears in the document and how rarely it appears in the rest of the corpora using zero mean normalization. Mean normalization is a method used to normalize a dataset by adjusting its values to have zero mean and equal variance. This method is utilized in ML and data analytics to preprocess data [45].

D. ML MODELS FOR LOG ANOMALY DETECTION

1) DECISION TREE (DT)

While a simple DT is not a black box model, it explains itself eloquently and it inherently reveals the important features in a simple prediction system. However, It is quite difficult for humans to fully grasp the reasoning behind every prediction in a complex DT. An explanations is require to help break down the complex logic in a simpler way. They can offer more details about why a specific prediction was made for a particular data point. This can be crucial for building trust in the model's decisions, especially in critical applications. These explanations can help uncover potential biases in the data or the model's training process. By understanding how the model uses features, it can identify if certain features are having an unfair or unintended influence on the results. Black boxes include RF, SVM, and other neural networks like CNN and Auto-Encoder. White-box models, such as Logistic Regression and Decision Trees, are naturally interpretable [46], [47]. The examination of classification and regression problems with a DT is an example of supervised learning [37]. Depending on predetermined input features, a recursive data partition is performed to conclude the dependent variable. The algorithm builds the tree until it reaches a stopping criterion.It stops selecting features to separate the data depending on metrics including information gain, gain ratio, and the Gini index. Following the decision path down, the tree can then be utilized to predict novel data inputs. We configured the DT model with the default parameter.

2) RANDOM FOREST (RF)

The RF utilizes a collection of DTs to produce more reliable and accurate forecasts [37]. Each decision tree in an RF model is unique because it is trained using a distinct sample of the data and a separate set of characteristics. The RF approach employs a bagging strategy during training in which numerous DTs are constructed, each trained using a bootstrap sample of the data from the entire dataset. While deciding how to divide a node in the DT, the algorithm randomly chooses some qualities to consider. When dealing with high-dimensional datasets or models with a large number of trees, RF models may show slower prediction generation than other techniques. Despite their reliability, RF models can nevertheless experience overfitting if not fine-tuned [38]. We employ a RF Classifier with the following features. It creates a collection of 10 DTs, each tree can reach a maximum depth of 10 levels. To divide a node within a tree, a minimum of 2 samples is necessary and we ensure consistency and reproducibility by utilizing a fixed random seed of 10.

3) GRADIENT BOOSTING (GB)

Gradient Boosting (GB) is an advanced ML technique popular for its remarkable predictive model and versatility as part of the ensemble learning family [48]. GB constructs a robust predictive model by combining the strengths of multiple simpler models, often decision trees. At its core, GB addresses the limitations of individual models by sequentially refining predictions, each step correcting the errors of its predecessors. The gradient concept guides this iterative process, highlighting how the model's performance can be enhanced. By iteratively building upon weak learners, It uncovers intricate patterns within data that might elude individual models, resulting in a powerful and finely-tuned predictive engine. We configured GB Classifier with the settings, to employ an ensemble of 10 DTs. There are no constraints on the maximum depth of each tree. Reproducibility and consistent results are guaranteed by using a fixed random seed of 10.

4) SXAD

The understand the workings of the SXAD framework we need to understand first, what are the Shapley values. The idea of Shapley values is derived from cooperative game theory, which allocate a player's share to the output of an entire game [49]. This assumes the presence of a cooperative game

Algorithm 1 SXAD Framework

Input : HDFS Dataset
Output: Key Features (Events)
Procedure *Data Parsing using Drain Parser and extracting features using TD-IDF*
Normalize *Features* using zero-mean;
Procedure *Log Anomaly Detection Models Training*
  Initialize Training and Testing Sets;
  Initialize Models;
  Models ← Models ∪ {Decision Tree (DT), Random Forest (RF), Gradient Boosting (GB)};
  for *each Model in Models* **do**
    Train Model using Training Data;
    Models ← Models ∪ {Trained Model};
  end
Procedure *Log Anomaly Detection Models Testing*
  Initialize Detected Anomalies;
  for *each Model in Models* **do**
    Apply Model to Testing Data to predict anomalies;
    Detected Anomalies ← Detected Anomalies ∪ {Predicted Anomalies};
  end
Procedure *Explain Anomalies using SHAP*
Initialize *Explanations* as an empty set;
for ***each*** *Anomaly in Detected Anomalies* **do**
  Use SHAP to explain the prediction of *Anomaly* for key feature;
  Visualize the explanations in the *Explanations* set;
end

in which a group of players collaborates to achieve a common objective. Shapley values represent the marginal constituency of each player towards the outcome. Shapley values are an idea acquired from the writing of insightful game hypotheses to evenly credit a player's commitment to the final product of a game [50]. These values catch the negligible commitment of every player to the outcome. By assuming, in our study, that each log record is a player in a game where the forecast determines the payoff, we may use this method to interpret the log anomalies. Numerically (1):

$$\phi_i = \sum_{x \to (D-i)} \frac{|x|!(|D| - |x| - 1)!}{D!} \left(y\left(x \cup \{i\}\right) - y(x)\right) \quad (1)$$

D = Feature set
x = Subset of features
i = Particular feature
y = Function that gives prediction

E. SHAP

SHAP is a ML interpretability approach that is model-independent [51]. The benefit of SHAP is that when employed in ML, it enables a computationally effective appraisal of Shapley values. It can be mathematically represented as (2):

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j + z'_j \quad (2)$$

g = model of explanation
z' $\in \{0, 1\}$
M = 1 means a presence of the feature and 0 means otherwise, z' represents the vector
M = Maximum size of coalition
ϕ_j = Attribution for a particular feature

LAD has various issues related to interpretability, accountability, transparency, and trust. The complex algorithms used for anomaly detection can make it difficult to interpret why certain events are flagged as anomalous, which can lead to a lack of transparency and trustworthiness in the system. Moreover, false positives and false negatives can further erode stakeholder trust in the effectiveness of the system [3].

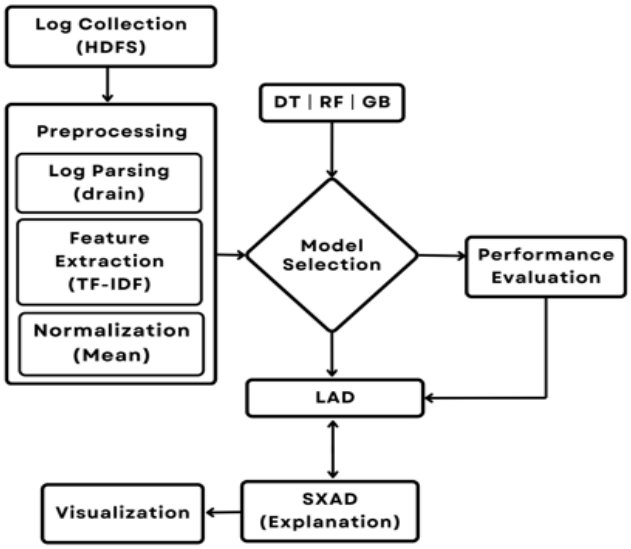

FIGURE 4. Flowchart of proposed SXAD framework.

XAI provide solutions to these challenges by increasing the interpretability and transparency of anomaly detection systems. XAI techniques like feature importance, rule-based systems, model interpretation, and human-in-the-loop approaches can help stakeholders understand the decision-making process of anomaly detection systems and increase their trust in the framework's capability to correctly distinguish and respond to anomalous events [47] and [48]. This paper proposes an SXAD Framework (Figure 2) using XAI-SHAP as a stepwise introduction about the working of the framework architecture is already defined earlier to better understand the methodology. After log collection, we parsed the HDFS dataset using Drain [39] to convert unstructured to structured logs, then applied TF-IDF [40]

and Mean Normalization [41] for feature extraction and normalization, and then applied various ML models (DT, RF, and GB) separately for Anomaly Detection, followed by the SHAP framework [17]. Analysis and results of the applied model with SHAP are discussed in the Result section. The working of the proposed SXAD architecture is depicted as a flowchart in Figure 4. In our work, Shapely values are the features of our dataset that contribute to the role towards model interpretability and explainability. The SHAP framework helps us to improve anomaly detection by providing explainability and increasing the interpretability and transparency of black box ML models [47]. The framework can identify which features are most important for detecting anomalies and help explain why certain events are being flagged as anomalous. By interpreting how the ML model is making decisions, stakeholders can better understand the system's decision-making process and increase their trust in its ability to accurately identify and respond to anomalous events [45]. The XAI-SHAP framework can also incorporate human professionals in the process of determining the course of action, further improving the accuracy and effectiveness of the system [49]. We provide the interpretable, explainable, and transparent analysis of the LAD framework using SHAP with numerous visualization explanations and interpretation techniques like Feature Importance, SHAP Summary Plot, and SHAP Force Plot. The detailed analysis of each step is mentioned below for easy understanding for the reader in the section mentioned below.

F. SHAP FEATURE IMPORTANCE

SHAP (Shapley Additive exPlanations) feature importance plot is a graphical representation of the effect of every feature on the model's predictions [52]. It is based on the Shapley values, which quantify the contribution of every feature.

The importance plot of the three (3) models by SHAP feature displays the most important features ranked by their absolute Shapley values, to the model output features to be displayed.

The SHAP feature importance plot is composed of a horizontal bar chart wherein each bar symbolizes the effect of a feature on the model's outcome. As depicted in Figure 5, 6, and 7 bar plots display SHAP values, illustrating the overall significance of each feature. This significance is determined by calculating the mean absolute value of each feature throughout the entirety of the dataset. The length of the bar represents the magnitude of the Shapley values. As the plot result shows the most important event (feature) is cf9b33dc and 797b9c47 in both DT and RF models. So we can further investigate the role of the most important event (feature) by analyzing the template of event ID.

G. SHAP SUMMERY PLOT

The SHAP (SHapley Additive exPlanations) summary plot is like a graph showing each feature's importance in a ML model. It does this by summing up how each feature

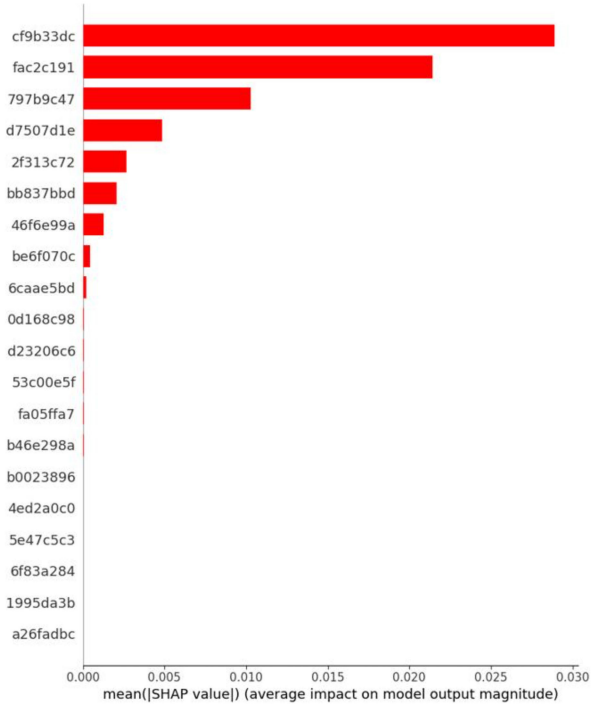

FIGURE 5. The importance plot by SHAP Feature using DT.

affects the model's predictions for the entire dataset. The idea behind this graph comes from Shapley values, which help determine the contribution of all the features to the model's prediction [52]. In the SHAP summary plot, every dot represents a single incident in the dataset, and they are all on a scatter plot. The X-axis shows the size of the SHAP value, which tells us how much a feature affects the model's prediction. On the Y-axis, we can see the name of the feature. The color of each dot shows the feature's value for that specific instance – blue means it is a low value, and red means a high value.

As shown in Figure 6, 7, and 8 for the SHAP summary plot using various implemented ML models, There is observable evidence that suggests a correlation between the value of a certain attribute and its influence on the prediction outcome.

From a visualizing and analysis perspective, the horizontal axis reflects the SHAP values associated with both high and low predictions. Meanwhile, the vertical axis, centered at zero (0.0), signifies no substantial effect on prediction outcomes. To clarify, a SHAP value of zero (0.0) suggests minimal influence on predictions, or values approaching zero indicate lower-quality predictions. Conversely, high-quality predictions, where SHAP values deviate significantly from zero, indicate either positive or negative correlations.

H. SHAP FORCE PLOT

The SHAP force plot is a type of graph that provides a detailed explanation of the output of an ML model for a single instance as for local explanation on HDFS dataset. Each instance represents a feature as a vertical bar in the plot, with the length

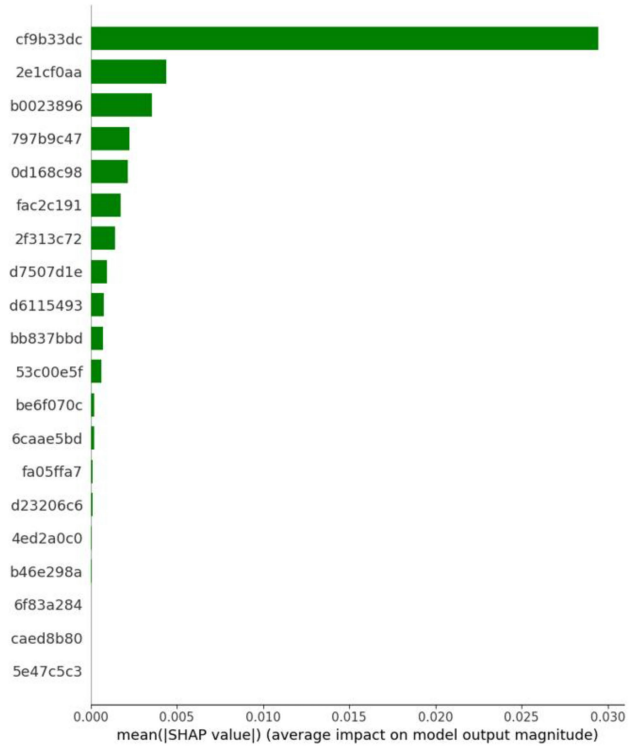

FIGURE 6. The importance plot by SHAP feature using RF.

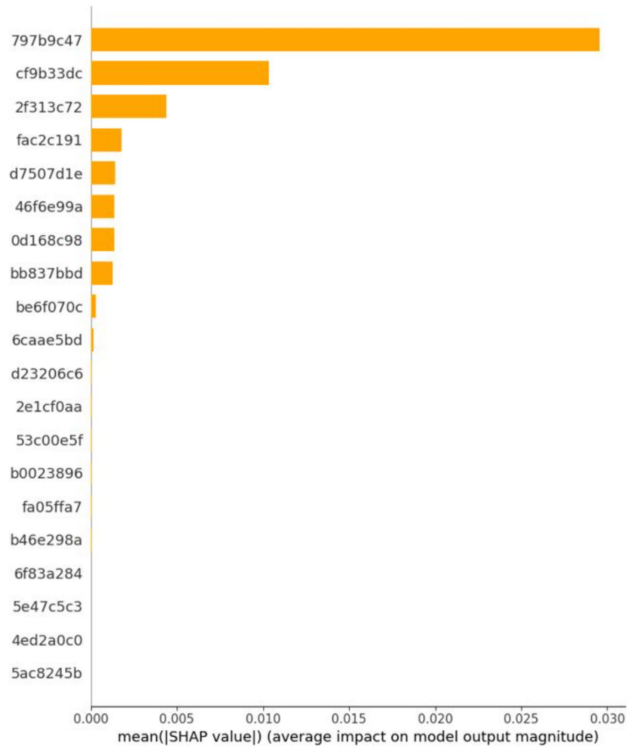

FIGURE 7. The importance plot by SHAP feature using GB.

of the bar exhibiting the feature's importance in determining the model's prediction. Every feature's contribution to the

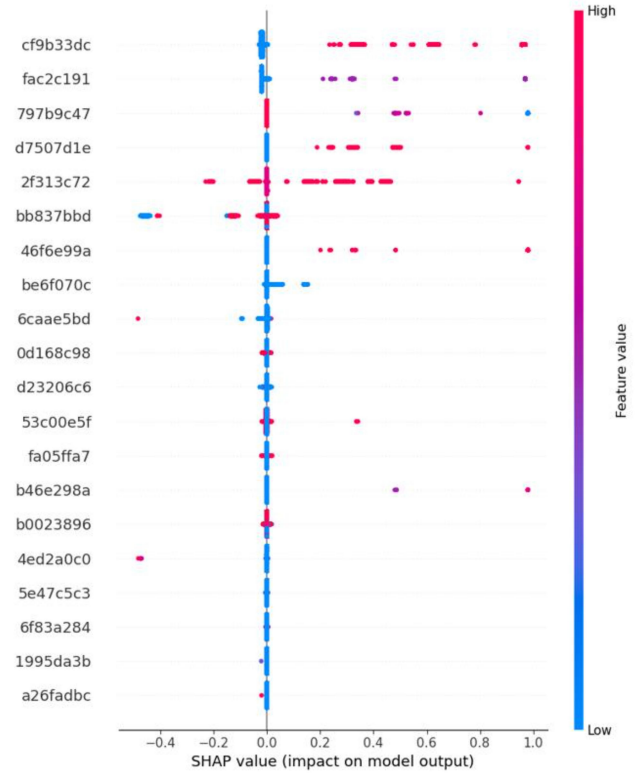

FIGURE 8. The summary plot by SHAP using DT.

prediction depicts the plot with an arrow pointing toward that influence [53]. The horizontal bar in the model represents the baseline value, corresponding to the average forecast throughout the entire dataset. The feature bars are assigned colors corresponding to their respective feature values, where blue shows a low value and red indicates a high value.

The SHAP force plot is an effective tool for evaluating the results of an ML model for a singular instance. Through a comprehensive examination of the plot, we can get insight into how each characteristic influences the forecast, enabling them to discern the most significant aspects of that specific occurrence. This tool can help users acquire a comprehensive insight into the model's action, enabling them to make well -informed judgments grounded in the predictions generated.

Figure 7 shows a particular log data insight using a forced plot of how various events contributed to model interpretability toward prediction. Red indicates features that increased the model's score, while blue indicates features that decreased the score.

IV. RESULT AND DISCUSSION

Given that LAD constitutes a binary classification task, we utilize precision, recall, and F1 score to assess its accuracy. In LAD precision measures the proportion of detected anomalous logs accurately identified as anomalies among all logs predicted as anomalies, while recall assesses the percentage of anomalies correctly identified by a model among all actual anomalies, with the F1 score representing the harmonic mean of precision and recall.

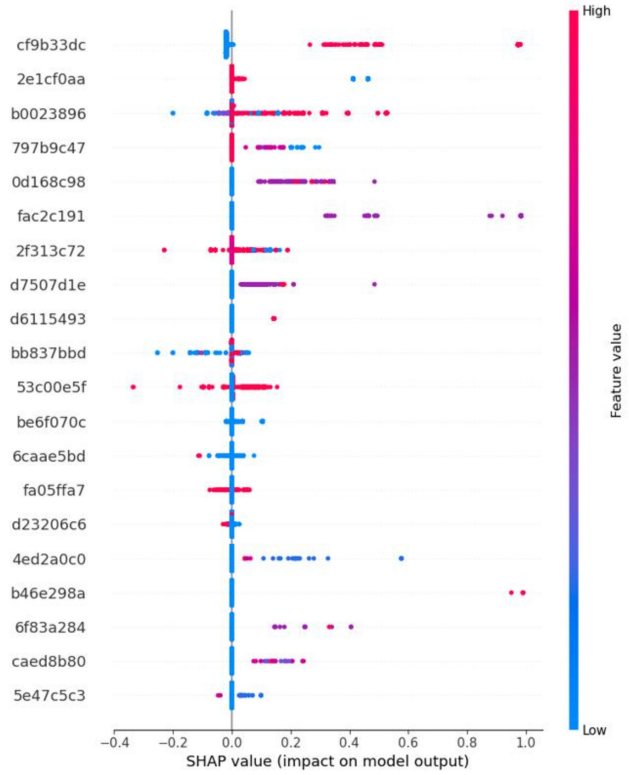

FIGURE 9. The summary plot by SHAP using RF.

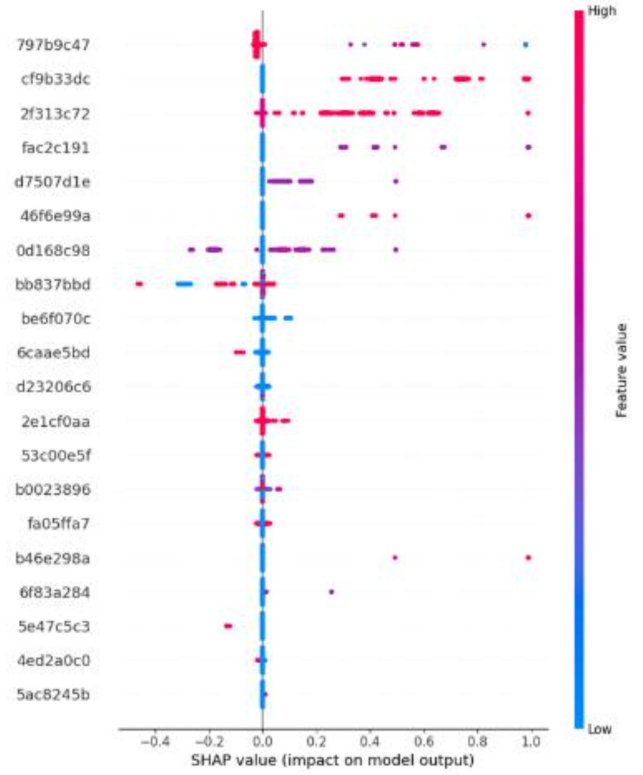

FIGURE 10. The importance plot by SHAP using GB.

A confusion matrix is a helpful instrument for analyzing the effectiveness of a classification model that is frequently employed in ML model evaluation techniques. The confusion matrix is a tabular representation of the model's predictions relative to the actual target variable values. It is commonly used to evaluate the model's accuracy, recall, and F1 score [3]. The confusion matrix comprises four(4) fundamental components: true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP).

True Positives (TP): The count of accurately classified positive observations. It represents the count of anomalies accurately identified by the model.

True Negatives (TN): The count of accurately anticipated negative observations.

False Positives (FP): The count of instances that were inaccurately classified as positive. It indicates the count of normal logs incorrectly predicted as anomalies by the model

False Negatives (FN): The count of instances that were inaccurately classified as negative. It signifies the count of anomalies that the model fails to detect.

By examining these four elements, we can derive several evaluation metrics:

A. ACCURACY

It measures the overall correctness of the model's predictions and is calculated as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

B. PRECISION

Precision focuses on the proportion of correctly predicted positive observations out of all positive predictions, and it is calculated as

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Precision indicates how reliable the positive predictions are.

C. RECALL

Recall is also known as true positive rate or sensitivity. It measures the proportion of correctly predicted positive observations out of all actual positive observations, and it is mathematically known as:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Recall shows the model's ability to find the positive instances correctly.

D. F1 SCORE

As a harmonious balance between precision and recall, the F1 Score gives a single measure of a model accuracy that covers both of these aspects. The calculation of F1 Score is as follows:

$$F1Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{6}$$

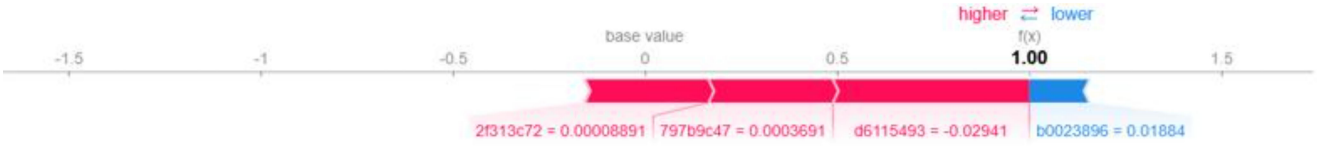

FIGURE 11. A view of force plot depicts the feature contribution at particular instances.

As we applied various ML models individually with the help of KernelSHAP method, which is a model agnostic XAI approach applies on benchmark HDFS system log. XAI-SHAP to deeply analyze the performance of the ML Model by leveraging the power of XAI-SHAP and its various interactive and interpretability visualization guidance to better understand the SXAD framework performance, like the role of key features contribution, its base value and impact towards model interpretability, trust and transparency. We use a 70:30 ratio dataset for the ML model and experiments are performed using the machine whose configurations are given below in Table 2.

TABLE 2. System details.

Processor	Core™ i7-9700 CPU @ 3.00 GHz
RAM	32 GB
OS	Windows 11 Pro 64 Bit

The results of using several different ML models are presented in Table 3. accuracy, precision, recall, and F1-measure are shown in percentage with four decimal digits All ML models are mentioned with performance along wit their accuracies.

TABLE 3. Model performances using various algorithms.

Model	Accuracy	Precision	Recall	F1-Measure	TN	FP	FN	TP
DT	99.9884	99.9710	99.9559	99.9971	167467	0	5	5047
RF	99.8516	99.8791	99.8791	99.8779	167464	3	7	5045
GB	99.9947	99.9947	99.8791	99.8779	167467	0	7	5045

All ML model efficiencies are remarkable but a comparison given values of the confusion matrix of all models shows approximately the same result in terms of True Negative (TN). The DT and GB model show approximately the same result among all models in terms of TP and False Positive FP values. In contrast, the traditional ML models are unable to provide an explanation, transparency, and feature contribution role toward model interpretability. Therefore, the XAI-SHAP model is developed to support both local and global model interpretability as well as transparency, trust, and explainability. The contribution of the top five events (features), in Table 2, of model interpretability towards positively detecting anomalies in the log. In addition, Table 3 shows the events (features) that contributed very little or did not affect model interpretability.

So at the end, we identify those key contributions of events (feature) towards positive model interpretability across all the models using a comparison of result in Table 1 and Table 2

TABLE 4. Table 4: Showing key five events (Features) effectively contribution towards model prediction.

Model	Events (Features)	Contribution
DT	cf9b33dc. fac2c191 797b9c47 d7507d1e 2f313c72	Positive
RF	cf9b33dc 2e1cf0aa boo23896 797b9c47 0d168c98	Positive
GB	797b9c47 cf9b33dc 2f313c72 fac2c191 d7507d1e	Positive

TABLE 5. Showing ineffective five key events (features) towards model predication.

Model	Events (Features)	Contribution
DT	bb837bbd 46f6e99a be6f070c 6caae5bd 0d168c98	Very Less / No Effect
RF	d6115493, bb837bbd 53c00e5c, be6fo7oc, 6caae5bd	Very Less / No Effect
GB	46f6e99a, 0d168c98, bb837bbd, be6fo7oc, 6caae5bd	Very Less / No Effect

results. The SXAD also facilitates the identification of those events (features) that can influence the model interpretation both positively and negatively (based on visualization results and data) in the future. So we provide these results, although they are very scarce, as we discover in our log data after detailed analysis of various visualization results as provided in Table 5.

We have further explored as the root-cause behind the system failure with the aid of event templates from system logs, and with the help of the SHAP Feature importance Plots. We determined the important event id as features contribution towards interoperability of LAD as shown in Table 7, we can examine and investigate this further as a step toward the

TABLE 6. Showing most influenced events (Features) contribution towards model predication.

S.No	Key Common Events
1	cf9b33dc
2	797b9c47
3	fac2c191
4	d7507dl3
5	2f313c72

TABLE 7. Showing most influenced events ID (features) with its corresponding event template.

Event ID	Event ID Template
cf9b33dc	Unexpected error trying to delete block blk_<*>. BlockInfo not found in volumeMap.
797b9c47	Received block blk_<*> of size <*> from /<*>
fac2c191	BLOCK* NameSystem.addStoredBlock: Redundant addStoredBlock request received for blk_<*> on <*>:50010 size <*>
d7507dl3	BLOCK* ask <*>:50010 to replicate blk_<*> to datanode(s) <*>:50010
2f313c72	BLOCK* NameSystem.addStoredBlock: blockMap updated: <*>:50010 is added to blk_<*> size <*>

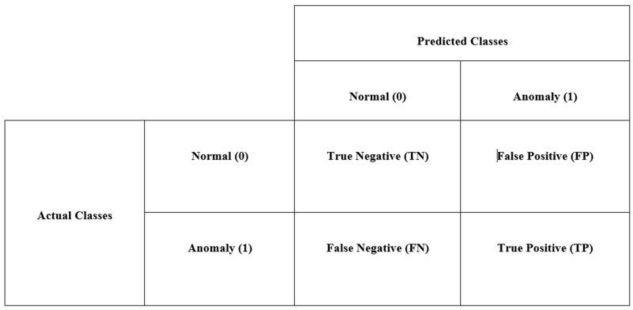

FIGURE 12. Confusion matrix.

explanation, transparency, and interpretation of log anomaly detection.

Our proposed framework and its results show all aspects of XAI-SHAP facilitation towards model interpretability, transparency, trust and explanation for LAD using the benchmark dataset HDFS System Log. As identification of key events contributing positively and negatively towards model interpretability, identification of such events has no role in model prediction. Furthermore, such critical events may influence interpretability or system performance in the future if taken in correlation. These key events can be further investigated in performing tasks such as troubleshooting, performance improvement, and debugging of the system.XAI improves LAD by explaining the causes of anomalous events. This reduces the amount of work required to handle false alarms while also increasing trust in the system by explaining why detection's happened. XAI also enables effective troubleshooting through the root-cause analysis and allows security specialists to configure the system to compliance their specified requirements. Overall, by leveraging log anomaly detection with XAI in Industry 5.0, organizations can achieve a significant reduction in system failures. This translates to increased efficiency, reduced downtime, and a more sustainable and human-centric approach to industrial maintenance.

V. CONCLUSION

This study explains SXAD utilization in detecting log anomalies using XAI, which is based on SHAP. An extensive HDFS dataset is utilized with a highly accurate Drain parser. Several ML models, such as DT, RF, and GB, are employed, showing remarkable accuracy levels. Furthermore, the suggested methodology incorporates various evaluation metrics, including precision, recall, and F1-Score, to facilitate a comprehensive examination of the performance measures. The objective of the presented method is to offer a solution that is more trustworthy, transparent, and interpretable in order to detect anomalies in logs. By comparing significant features, we determine the most impactful events (features) have on enhancing interpretability in all models. The proposed SXAD framework makes it easier to recognize occurrences (features) that have the potential to influence model interpretation in either a favorable or unfavorable way. A more in-depth investigation of the underlying causes of failures, a knowledge of the roles that feature contributions play in anomalies, and interpretation through a variety of visualization techniques are all made possible by the proposed framework. Our efforts also highlight the significance of applying XAI to detect anomalies in system log which might causes large-scale system failure, in order to realization the aim of Industry 5.0. Our contribution are novel in the area of LAD because no earlier work has been done, establishing the way for future research and progress in the field of LAD.

VI. FUTURE WORK

Machine learning model need fairness. Our future work will extend our contribution to reduction in fairness and biasness in XAI models as is critical for ethical consideration.

REFERENCES

[1] J. Breier and J. Branišová, "Anomaly detection from log files using data mining techniques," in *Information Science and Applications*. Springer, 2015, pp. 449–457.
[2] Z. Rehman, N. Tariq, S. A. Moqurrab, J. Yoo, and G. Srivastava, "Machine learning and Internet of Things applications in enterprise architectures: Solutions, challenges, and open issues," *Exp. Syst.*, vol. 41, no. 1, Jan. 2024, Art. no. e13467.
[3] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," *Mach. Learn. Appl.*, vol. 12, Jun. 2023, Art. no. 100470.
[4] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1285–1298.
[5] J. Svacina, J. Raffety, C. Woodahl, B. Stone, T. Cerny, M. Bures, D. Shin, K. Frajtak, and P. Tisnovsky, "On vulnerability and security log analysis: A systematic literature review on recent trends," in *Proc. Int. Conf. Res. Adapt. Convergent Syst.*, 2020, pp. 175–180.
[6] S. He, P. He, Z. Chen, T. Yang, Y. Su, and M. R. Lyu, "A survey on automated log analysis for reliability engineering," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–37, Jul. 2022.
[7] V. Hassija, V. Chamola, A. Mahapatra, A. Singal, D. Goel, K. Huang, S. Scardapane, I. Spinelli, M. Mahmud, and A. Hussain, "Interpreting black-box models: A review on explainable artificial intelligence," *Cognit. Comput.*, vol. 16, no. 1, pp. 45–74, Jan. 2024.

[8] V. Chamola, V. Hassija, A. R. Sulthana, D. Ghosh, D. Dhingra, and B. Sikdar, "A review of trustworthy and explainable artificial intelligence (XAI)," *IEEE Access*, vol. 11, pp. 78994–79015, 2023.

[9] P. J. Phillips, C. A. Hahn, P. C. Fontana, D. A. Broniatowski, and M. A. Przybocki, "Four principles of explainable artificial intelligence," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 8312, 2020, vol. 18.

[10] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, "Explainable intrusion detection for cyber defences in the Internet of Things: Opportunities and solutions," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1775–1807, 3rd Quart., 2023.

[11] L. Chen, Y. Li, W. Silamu, Q. Li, S. Ge, and F.-Y. Wang, "Smart mining with autonomous driving in industry 5.0: Architectures, platforms, operating systems, foundation models, and applications," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 3, pp. 4383–4393, Mar. 2024.

[12] P. Yan, A. Abdulkadir, P.-P. Luley, M. Rosenthal, G. A. Schatte, B. F. Grewe, and T. Stadelmann, "A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods, applications, and directions," *IEEE Access*, vol. 12, pp. 3768–3789, 2024.

[13] N. Jeffrey, Q. Tan, and J. R. Villar, "A review of anomaly detection strategies to detect threats to cyber-physical systems," *Electronics*, vol. 12, no. 15, p. 3283, Jul. 2023.

[14] A.-R. Al-Ghuwairi, Y. Sharrab, D. Al-Fraihat, M. AlElaimat, A. Alsarhan, and A. Algarni, "Intrusion detection in cloud computing based on time series anomalies utilizing machine learning," *J. Cloud Comput.*, vol. 12, no. 1, p. 127, Aug. 2023.

[15] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[16] D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen, "Explainable artificial intelligence: A comprehensive review," *Artif. Intell. Rev.*, vol. 55, no. 5, pp. 3503–3568, Jun. 2022.

[17] Z. Li, Y. Zhu, and M. Van Leeuwen, "A survey on explainable anomaly detection," *ACM Trans. Knowl. Discovery Data*, vol. 18, no. 1, pp. 1–54, Jan. 2024.

[18] S. Du and J. Cao, "Behavioral anomaly detection approach based on log monitoring," in *Proc. Int. Conf. Behav., Econ. Socio-Cultural Comput. (BESC)*, 2015, pp. 188–194.

[19] C. Bertero, M. Roy, C. Sauvanaud, and G. Tredan, "Experience report: Log mining using natural language processing and application to anomaly detection," in *Proc. IEEE 28th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2017, pp. 351–360.

[20] S. Lu, X. Wei, Y. Li, and L. Wang, "Detecting anomaly in big data system logs using convolutional neural network," in *Proc. IEEE 16th Intl Conf Dependable, Autonomic Secure Comput., 16th Intl Conf Pervasive Intell. Comput., 4th Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congress(DASC/PiCom/DataCom/CyberSciTech)*, Aug. 2018, pp. 151–158.

[21] M. Wang, L. Xu, and L. Guo, "Anomaly detection of system logs based on natural language processing and deep learning," in *Proc. 4th Int. Conf. Frontiers Signal Process. (ICFSP)*, Sep. 2018, pp. 140–144.

[22] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, and R. Zhou, "LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4739–4745.

[23] W. Meng, Y. Liu, S. Zhang, D. Pei, H. Dong, L. Song, and X. Luo, "Device-agnostic log anomaly classification with partial labels," in *Proc. IEEE/ACM 26th Int. Symp. Quality Service (IWQoS)*, Jun. 2018, pp. 1–6.

[24] N. Gurumdimma, A. Jhumka, M. Liakata, E. Chuah, and J. Browne, "CRUDE: Combining resource usage data and error logs for accurate error detection in large-scale distributed systems," in *Proc. IEEE 35th Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2016, pp. 51–60.

[25] T. Jia, L. Yang, P. Chen, Y. Li, F. Meng, and J. Xu, "LogSed: Anomaly diagnosis through mining time-weighted control flow graph in logs," in *Proc. IEEE 10th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2017, pp. 447–455.

[26] J. Wang, Y. Tang, S. He, C. Zhao, P. K. Sharma, O. Alfarraj, and A. Tolba, "LogEvent2vec: LogEvent-to-vector based anomaly detection for large-scale logs in Internet of Things," *Sensors*, vol. 20, no. 9, p. 2451, Apr. 2020.

[27] S. He, J. Zhu, P. He, and M. R. Lyu, "Experience report: System log analysis for anomaly detection," in *Proc. IEEE 27th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2016, pp. 207–218.

[28] X. Xie, Z. Jin, J. Wang, L. Yang, Y. Lu, and T. Li, "Confidence guided anomaly detection model for anti-concept drift in dynamic logs," *J. Netw. Comput. Appl.*, vol. 162, Jul. 2020, Art. no. 102659.

[29] T. Jia, P. Chen, L. Yang, Y. Li, F. Meng, and J. Xu, "An approach for anomaly diagnosis based on hybrid graph model with logs for distributed services," in *Proc. IEEE Int. Conf. Web Services (ICWS*, 2017, pp. 25–32.

[30] R. Vaarandi and M. Pihelgas, "LogCluster—A data clustering and pattern mining algorithm for event logs," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2015, pp. 1–7.

[31] S. He, Q. Lin, J.-G. Lou, H. Zhang, M. R. Lyu, and D. Zhang, "Identifying impactful service system problems via log analysis," in *Proc. 26th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2018, pp. 60–70.

[32] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, J. Chen, X. He, R. Yao, J.-G. Lou, M. Chintalapati, F. Shen, and D. Zhang, "Robust log-based anomaly detection on unstable log data," in *Proc. 27th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Aug. 2019, pp. 807–817.

[33] D. Li, J. Zhang, X. Zhang, F. Lin, C. Wang, and L. Chang, "LogPS: A robust log sequential anomaly detection approach based on natural language processing," in *Proc. IEEE 22nd Int. Conf. Commun. Technol. (ICCT)*, Nov. 2022, pp. 1400–1405.

[34] B. Xia, J. Yin, J. Xu, and Y. Li, "LogGAN: A sequence-based generative adversarial network for anomaly detection based on system logs," in *Proc. 2nd Int. Conf. SciSec Sci. Cyber Secur.*, vol. 2, 2019, pp. 61–76.

[35] R. Chen, S. Zhang, D. Li, Y. Zhang, F. Guo, W. Meng, D. Pei, Y. Zhang, X. Chen, and Y. Liu, "LogTransfer: Cross-system log anomaly detection for software systems with transfer learning," in *Proc. IEEE 31st Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2020, pp. 37–47.

[36] B. Li, S. Ma, R. Deng, K. R. Choo, and J. Yang, "Federated anomaly detection on system logs for the Internet of Things: A customizable and communication-efficient approach," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 2, pp. 1705–1716, Jun. 2022.

[37] P. Raut, A. Mishra, S. Rao, S. Kawoor, S. Shelke, M. Deore, and V. Kumar, "Review on log-based anomaly detection techniques," in *Proc. 2nd Int. Conf. Sustain. Exp. Syst. (ICSES)*. Springer, 2022, pp. 893–906.

[38] G.-H. Yang, G.-Y. Zhong, L.-Y. Wang, Z.-G. Xie, and J.-C. Li, "A hybrid forecasting framework based on MCS and machine learning for higher dimensional and unbalanced systems," *Phys. A, Stat. Mech. Appl.*, vol. 637, Mar. 2024, Art. no. 129612.

[39] S. He, J. Zhu, P. He, and M. R. Lyu, "Loghub: A large collection of system log datasets towards automated log analytics," 2008, *arXiv:2008.06448*.

[40] W. Xu, L. Huang, A. Fox, D. Patterson, and M. Jordan, "Largescale system problem detection by mining console logs," in *Proc. SOSP*, 2009.

[41] T. Xiao, Z. Quan, Z.-J. Wang, Y. Le, Y. Du, X. Liao, K. Li, and K. Li, "Loader: A log anomaly detector based on transformer," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3479–3492, Sep. 2023.

[42] X. Sun, Y. He, D. Wu, and J. Z. Huang, "Survey of distributed computing frameworks for supporting big data analysis," *Big Data Mining Analytics*, vol. 6, no. 2, pp. 154–169, Jun. 2023.

[43] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 33–40.

[44] S. Pasarate and R. Shedge, "Comparative study of feature extraction techniques used in sentiment analysis," in *Proc. Int. Conf. Innov. Challenges Cyber Secur. (ICICCS-INBUSH)*. Springer, Feb. 2016, pp. 475–486.

[45] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 105524.

[46] T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf, "Principles of explanatory debugging to personalize interactive machine learning," in *Proc. 20th Int. Conf. Intell. User Interfaces*, Mar. 2015, pp. 126–137.

[47] C. Molnar, *Interpretable Machine Learning*. Leanpub, 2022.

[48] J. Frery, A. Habrard, M. Sebban, O. Caelen, and L. He-Guelton, "Efficient top rank optimization with gradient boosting for supervised anomaly detection," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML PKDD)*, Skopje, Macedonia. Springer, Sep. 2017, pp. 20–35.

[49] Z. Li, "Extracting spatial effects from machine learning model using local interpretation method: An example of SHAP and XGBoost," *Comput., Environ. Urban Syst.*, vol. 96, Sep. 2022, Art. no. 101845.

[50] A. Messalas, Y. Kanellopoulos, and C. Makris, "Model-agnostic interpretability with Shapley values," in *Proc. 10th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2019, pp. 1–7.

[51] K. Aas, M. Jullum, and A. Løland, "Explaining individual predictions when features are dependent: More accurate approximations to Shapley values," *Artif. Intell.*, vol. 298, Sep. 2021, Art. no. 103502.
[52] H. Chen, I. C. Covert, S. M. Lundberg, and S.-I. Lee, "Algorithms to estimate Shapley value feature attributions," *Nature Mach. Intell.*, vol. 5, no. 6, pp. 590–601, May 2023.
[53] T. Speith, "A review of taxonomies of explainable artificial intelligence (XAI) methods," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, Jun. 2022, pp. 2239–2250.

KASHIF ALAM is currently pursuing the Ph.D. degree with the Department of Computer Science, FCAI, Air University, Islamabad. He is an experienced professional with a long-standing background in teaching and training. He demonstrated proficiency in actively engaging in many national and international financed initiatives, demonstrating a strong dedication to research and education. His research interests include network security, computer forensics, and machine learning.

KASHIF KIFAYAT received the Ph.D. degree in cyber security from Liverpool John Moores University, Liverpool, U.K., in 2008. He is currently a Professor with the Department of Cyber Security, FCAI, Air University, Islamabad, Pakistan, and the Director of the National Centre for Cyber Security (NCCS). Besides publishing more than 100 research articles. His research interests include the security of complex systems, intrusion detection systems, digital forensics, privacy-preserving data aggregation, computer forensics, and cryptography. He has also played a key role in many funded research and development projects related to his research topics. He is a member of various research journal editorial boards.

GABRIEL AVELINO SAMPEDRO (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Mapúa University, Manila, Philippines, in 2018, and the Ph.D. degree in IT convergence engineering from the Kumoh National Institute of Technology, Gumi-si, South Korea, in 2023. Currently, he is an Assistant Professor with the Faculty of Information and Communication Studies, University of the Philippines Open University, and a Researcher with the Center for Computational Imaging and Visual Innovations, De La Salle University. His research interests include real-time systems, embedded systems, robotics, and biomedical engineering.

VINCENT KAROVIČ JR. is currently the Vice-Dean of IT of the Faculty of Management, Comenius University Bratislava. He is a Distinguished Academician and a Researcher, published a large number of articles in internationally renewed journals. His research interests include computer networks, information security, virtualization, information resources, and cloud computing.

TARIQ NAEEM is currently an Assistant Professor with the Department of Computer Science, Faculty of Computing and Artificial Intelligence, Air University, Islamabad, Pakistan, where he has been, since 2016. His research interests include NLP, machine learning, deep learning, and mission-critical environments.