Anomaly Detection
Universal Model

Mikhail Nazarenko©
mikhail.nazarenko@gmail.com

Updated
October 30, 2024

Contents

Abstract

In this paper, we present a comprehensive model for anomaly detection, designed to adapt to the dynamics and contextual variations in monitored systems. The proposed model combines several key components, including dynamic thresholding, metric hierarchy, correlation analysis, and time series forecasting, to enhance both the precision and adaptability of anomaly detection processes. Dynamic thresholds are established based on historical data and current context, allowing the model to adjust to varying operational conditions and reduce false positives.

The model also incorporates a hierarchical structure for metrics, categorizing them into critical, warning, and informational levels, which allows for prioritized monitoring and response. Correlation analysis between metrics is used to identify interdependencies, further refining the accuracy of anomaly detection. Time series forecasting enables the system to anticipate potential deviations and take proactive measures.

The formalization of dynamic thresholds, metric hierarchies, and forecasting functions, along with adaptive filtering rules, ensures the model is both robust and capable of maintaining stability in complex environments. The integration of these elements creates a flexible, highly reliable framework for real-time anomaly detection and monitoring.

Keywords: Anomaly Detection, Dynamic Thresholding, Metric Hierarchy, Correlation Analysis, Time Series Forecasting, Real-Time Monitoring.

Part I

Universal Model

Chapter 1

Complete Universal Anomaly Detection System Model

1.1 Formalization of the General System Model

This anomaly detection system model is based on dynamic thresholds, filtering rules, correlation analysis, time-series-based forecasting, and metric hierarchy. The improved model incorporates adaptive parameters for more accurate and flexible real-time anomaly detection.

1.2 Dynamic Thresholds $M_{\text{dynamic\_threshold}}$

1.2.1 Defining Dynamic Thresholds

Let $M_{\text{dynamic\_threshold}}$ be the set of dynamic thresholds, defining flexible boundaries for each metric based on its historical behavior and current context. The threshold for each metric m_i is calculated as:

$$M_{\text{dynamic\_threshold}} = \{T_i \mid T_i = f(\text{historical\_data}(m_i), \text{context}(m_i))\} \tag{1.1}$$

where function f considers components such as seasonality and trends.

1.2.2 Contextual Adaptation of Thresholds

Contextual conditions $C(T_i)$ include parameters like time of day, day of the week, and seasonal fluctuations:

$$T_i = f(\text{historical\_data}(m_i), C(T_i)) \tag{1.2}$$

Context adaptation allows thresholds to adjust based on current conditions, enhancing detection accuracy.

1.3 False Positive Filtering F

1.3.1 Filtering Rules

Let F be the set of filtering rules designed to minimize false positives:

$$F = \{f_k(m) \mid f_k(m) - \text{a condition for filtering out minor anomalies}\} \tag{1.3}$$

Each filtering rule dynamically adjusts based on the current data, improving resistance to noise.

1.3.2 Adaptive Filtering Rules Based on Correlations

We add adaptive rules that consider correlations between metrics to prevent false positives:

$$f_{\text{correlated}}(m_i, m_j) = \begin{cases} 1, & \text{if } r_{ij} > \gamma \\ 0, & \text{otherwise} \end{cases} \tag{1.4}$$

where r_{ij} is the correlation coefficient between metrics m_i and m_j, and γ is the significant correlation threshold.

1.4 Correlation Analysis $C_{\text{correlation\_analysis}}$

1.4.1 Correlation Coefficients

To analyze linear dependencies, we use Pearson's correlation coefficient r_{ij}:

$$r_{ij} = \frac{\sum_{k=1}^{n}(m_{i,k} - \bar{M}_i)(m_{j,k} - \bar{M}_j)}{\sqrt{\sum_{k=1}^{n}(m_{i,k} - \bar{M}_i)^2} \cdot \sqrt{\sum_{k=1}^{n}(m_{j,k} - \bar{M}_j)^2}} \tag{1.5}$$

For non-linear dependencies, Spearman's rank coefficient ρ_{ij} is used, defined by ranking values:

$$\rho_{ij} = 1 - \frac{6\sum_{k=1}^{n}(R(m_{i,k}) - R(m_{j,k}))^2}{n(n^2 - 1)} \tag{1.6}$$

1.4.2 Lagged Correlation Analysis

Lagged analysis is introduced to reveal temporal dependencies between metrics:

$$r_{ij}(\tau) = \frac{\sum_{k=1}^{n-\tau}(m_{i,k} - \bar{M}_i)(m_{j,k+\tau} - \bar{M}_j)}{\sqrt{\sum_{k=1}^{n-\tau}(m_{i,k} - \bar{M}_i)^2} \cdot \sqrt{\sum_{k=1}^{n-\tau}(m_{j,k+\tau} - \bar{M}_j)^2}} \tag{1.7}$$

where τ is the time lag.

1.5 Time-Series-Based Forecasting $P_{\text{prediction}}$

1.5.1 Defining the Prediction Model

The forecasting model g is based on historical data and applied to each time series $m_i(t)$:

$$\hat{m}_i(t + \Delta t) = a_0 + \sum_{k=1}^{p} a_k m_i(t - k) + \epsilon \tag{1.8}$$

where a_k are model coefficients, and ϵ is random noise.

1.5.2 Adaptive Model Re-training

Model parameters are periodically updated:

$$\theta_{t+1} = \theta_t + \alpha \cdot \nabla_\theta L(\theta, M_{\text{time\_series}}) \tag{1.9}$$

where $L(\theta, M_{\text{time\_series}})$ is a loss function minimizing the difference between actual and predicted values.

1.6 Metric Hierarchy $H_{\text{metric\_hierarchy}}$

1.6.1 Metric Criticality Levels

The metric hierarchy is divided into three levels of criticality:

$$H_{\text{metric\_hierarchy}} = \text{group}(\{M_{\text{critical}}, M_{\text{warning}}, M_{\text{info}}\}) \tag{1.10}$$

Each level is assigned a separate weighting coefficient for the aggregated assessment of system status.

1.6.2 Dynamic Reclassification of Metrics

Metrics can move between criticality levels depending on the context, formalized as:

$$\text{If } m_i \in M_{\text{info}} \text{ and } |m_i - \mu(m_i)| > \delta, \text{ then } m_i \to M_{\text{critical}} \tag{1.11}$$

where δ is the threshold for transition.

1.7 Conclusion

The updated anomaly detection system model provides high adaptability and accuracy, integrating dynamic thresholds, enhanced filtering, correlation analysis, forecasting, and

a flexible metric hierarchy. All components work in concert to detect anomalies in real-time, enhancing the system's reliability and resilience.

Part II

Anomaly Detection

Chapter 2

Mathematical Formalization of the Anomaly Detection System

2.1 Primary System Parameters

Consider the following key parameters of the anomaly detection system:

- M — the set of metrics that define system parameters, such as video quality, response time, network traffic, etc.
- T — the set of anomaly triggers, which establish the conditions for metrics to deviate from normal ranges.
- D — the set of data sources, including external monitoring systems and internal databases.
- F — the filtering rules, which minimize the number of false positives.

2.2 Formalization of Metrics and Anomalies

Let $S \subseteq M$ be a subset of metrics used for anomaly detection. Each anomaly is defined when the current metric values exceed preset thresholds:

$$f(S_i, T_j) = \begin{cases} 1, & \text{if } T_j(S_i) > \text{threshold;} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

where $T_j(S_i)$ represents the metric S_i at a given time as defined by trigger T_j. An anomaly is recorded if $T_j(S_i)$ exceeds the established threshold.

2.3 Dynamic Thresholds for Contextual Adaptation

To enhance system adaptability, dynamic thresholds incorporate historical data and context. Let $M_{\text{dynamic\_threshold}}$ be the set of dynamically defined thresholds, each updated based on historical information and external factors:

$$M_{\text{dynamic\_threshold}} = \{T_i \mid T_i = f(\text{historical\_data}, \text{context})\}, \tag{2.2}$$

where f is a function that uses context and historical data to update thresholds, thus detecting deviations more accurately.

2.4 Anomaly Prediction Model

A time-series model is used to predict potential deviations. Let $P_{\text{prediction}}$ be the prediction subsystem, which calculates the likelihood of anomalies based on time series data:

$$P_{\text{prediction}} = g(M_{\text{time\_series}}), \tag{2.3}$$

where g is a time-series analysis function that examines sequential metric values to predict the probability of future anomalies. This function enables the system to respond proactively to potential threats.

2.5 Hierarchical Structure of Metrics

To manage metrics efficiently and reduce analysis complexity, a hierarchical structure $H_{\text{metric\_hierarchy}}$ is introduced, grouping metrics by levels of importance:

$$H_{\text{metric\_hierarchy}} = \text{group}(\{M_{\text{critical}}, M_{\text{warning}}, M_{\text{info}}\}), \tag{2.4}$$

where M_{critical} are critical metrics that directly impact core functionality, M_{warning} are warning metrics indicating potential issues, and M_{info} are informational metrics that provide general system status.

2.6 Correlation Analysis of Metrics

To improve detection accuracy, correlation analysis between metrics is used. Let $C_{\text{correlation\_analysis}}$ be a subset of correlated metrics. Each pair of metrics M_i and M_j has a degree of correlation $h(M_i, M_j)$:

$$C_{\text{correlation\_analysis}} = h(M_i, M_j), \tag{2.5}$$

where h is a function that determines the degree of correlation between metrics. High correlation between two metrics accounts for interdependencies, improving anomaly detection accuracy.

2.7 False Positive Filtering Rules

To minimize false positives, a subset of rules F is defined to exclude metric deviations within permissible limits. Filtering rules are defined by the following conditions:

$$F(S_i) = \begin{cases} 0, & \text{if } |S_i - \text{median}| < \Delta; \\ 0, & \text{if } S_i \in N_{\text{short-term}}; \\ 0, & \text{if anomaly age} > T_{\text{max}}; \\ 1, & \text{otherwise.} \end{cases} \tag{2.6}$$

where Δ is the permissible deviation around the metric's median, $N_{\text{short-term}}$ denotes short-term anomalies that do not require response, and T_{max} is the maximum permissible anomaly age.

2.8 Optimization and Load Distribution

For increased performance under high load, the system distributes requests across clusters. Let K represent the set of clusters handling requests with varying priorities. A request is assigned to a specific cluster based on:

$$C(R) = \begin{cases} k_{\text{fast}}, & \text{if request processing speed is high;} \\ k_{\text{standard}}, & \text{if request processing speed is standard;} \\ k_{\text{slow}}, & \text{otherwise.} \end{cases} \tag{2.7}$$

where R denotes the request type, and $k_{\text{fast}}, k_{\text{standard}}, k_{\text{slow}}$ are clusters ensuring different processing speeds depending on request priority.

2.9 Conclusion

Thus, the resulting mathematical model of the anomaly detection system integrates dynamic thresholds, prediction, metric hierarchy, correlation analysis, and false positive filtering. This model enhances the accuracy and reliability of anomaly detection, minimizing false signals and adapting to changing conditions.

2.10 System Metric Set Formalization

Let M denote the set of metrics characterizing system parameters and determining its quality and performance. Key parameters include video quality, response time, network traffic, and other indicators. Each metric in this set can be described in terms of its values, dynamics, and impact on the overall system state. To formalize the metric system, we introduce the following definitions and structures.

2.10.1 Set of Metrics and Their Description

Define the set of metrics M as the collection of all measurable parameters characterizing system operation:

$$M = \{m_1, m_2, \ldots, m_n\} \tag{2.8}$$

where m_i represents an individual metric, such as video quality, service response time, network traffic, and other parameters. Each metric m_i can be represented as a function of time and other variables:

$$m_i = f_i(t, X) \tag{2.9}$$

where t denotes time, and X represents a set of external factors affecting the value of metric m_i.

2.10.2 Metric Classification

For efficient analysis and management, metrics are categorized by their importance and the type of parameter they measure. Define the classification C for the set of metrics M:

$$C(M) = \{M_{\text{critical}}, M_{\text{warning}}, M_{\text{informational}}\} \tag{2.10}$$

where M_{critical} represents critical metrics directly impacting system performance and functionality; M_{warning} indicates warning metrics that signal potential issues; and $M_{\text{informational}}$ includes informational metrics that monitor system state without significantly affecting its operation.

2.10.3 Membership Function and Metric Weight

Each metric $m_i \in M$ is assigned a weight w_i, reflecting its significance in the overall system, and a membership function $\mu(m_i)$, which determines its importance level based on the system's current conditions:

$$w(m_i) = \begin{cases} 1, & \text{if } m_i \in M_{\text{critical}} \\ 0.5, & \text{if } m_i \in M_{\text{warning}} \\ 0, & \text{if } m_i \in M_{\text{informational}} \end{cases} \tag{2.11}$$

$$\mu(m_i, C) = \frac{1}{1 + e^{-\alpha(x-\beta)}} \tag{2.12}$$

where α and β are parameters defining the shape of the membership function $\mu(m_i, C)$, adjusted according to the criticality and operational conditions of the system.

2.10.4 Temporal Dynamics and Metric Variation

Each metric m_i is also characterized by its temporal dynamics, described through its rate of change over time:

$$\frac{dm_i}{dt} = g_i(m_i, t, X) \tag{2.13}$$

where g_i is a function defining the rate of change of metric m_i with respect to time and external factors X. This allows tracking how critical metrics evolve over time and how quickly the system responds to external influences.

2.10.5 Impact of Metrics on Overall System State

The system state S depends on the aggregate value of all metrics. Let S be a function of all metrics M, then the overall system state can be defined as:

$$S = H(M) = H(m_1, m_2, \ldots, m_n) \tag{2.14}$$

where H is an aggregating function that takes into account the weights and current values of each metric m_i. The function H could, for example, be a weighted average or a more complex function reflecting each metric's importance to the system.

2.10.6 Conflicts Between Metrics and Balance Conditions

Certain metrics may conflict with one another, described by a conflict function R. Let $R \subseteq M \times M$ be the set of conflicting metric pairs, where each pair (m_i, m_j) belongs to R if there exist conditions $c_k \in C$ under which m_i and m_j have opposing values:

$$\exists c_k \in C : f(m_i, c_k) = -f(m_j, c_k) \tag{2.15}$$

To resolve conflicts, balance conditions are introduced so that metric values m_i and m_j are aligned:

$$\forall (m_i, m_j) \in R \Rightarrow (\exists c_m : f(m_i, c_m) = f(m_j, c_m)) \tag{2.16}$$

where c_m is a specific condition under which the conflict between metrics is resolved, allowing the system to maintain a stable state S.

2.10.7 Conclusion

The formalized set of metrics M, including classification, weights, temporal dynamics, system state impact, and conflict conditions, provides a basis for system analysis and management in real operational settings. This structure enables adaptation to changing conditions and supports stable system performance through dynamic analysis and inter-metric relationships.

2.11 Formalization of the Set of Anomaly Threshold Triggers

Let T be the set of triggers that establish threshold conditions for metric deviations from normal ranges, used to identify system anomalies. These triggers activate when metric values exceed or fall below critical thresholds, signaling deviations in system behavior. The definition and structure of triggers include conditions, weight values, and temporal parameters for accurate monitoring and timely anomaly detection.

2.11.1 Set of Triggers and Their Definition

Define the set of triggers T as a collection of conditions that activate anomaly signals:

$$T = \{t_1, t_2, \ldots, t_k\} \tag{2.17}$$

where t_i is an individual trigger responsible for monitoring specific metric values and their deviations from set thresholds. Each trigger t_i can be represented as a logical function of the metric m_i:

$$t_i = \begin{cases} 1, & \text{if } m_i \notin [L_i, U_i] \\ 0, & \text{otherwise} \end{cases} \tag{2.18}$$

where L_i and U_i are the lower and upper threshold values for metric m_i. When a metric value is outside these bounds, trigger t_i activates, indicating an anomaly.

2.11.2 Classification of Triggers by Criticality

For accurate monitoring, triggers are classified according to their criticality and significance:

$$C(T) = \{T_{\text{critical}}, T_{\text{warning}}, T_{\text{informational}}\} \tag{2.19}$$

where T_{critical} includes triggers indicating critical deviations requiring immediate response; T_{warning} includes warning triggers that signal potential risks; and $T_{\text{informational}}$ includes informational triggers that monitor the system state but are non-critical.

2.11.3 Membership Function for Trigger Criticality Level

Each trigger $t_i \in T$ is assigned a membership function $\mu(t_i, C)$, which indicates the likelihood of criticality for the current deviation and assesses the need for action:

$$\mu(t_i, C) = \frac{1}{1 + e^{-\alpha(x-\beta)}} \tag{2.20}$$

where α and β are parameters of the membership function, depending on the criticality level and nature of the trigger. The membership function value indicates the response requirement upon trigger activation.

2.11.4 Temporal Parameters for Trigger Activation

Each trigger t_i is assigned temporal parameters to determine how long a metric value must remain outside thresholds to activate an anomaly signal:

$$\tau_i = \int_{t_0}^{t_1} f(m_i)\, dt \tag{2.21}$$

where τ_i is the time interval during which metric m_i must remain outside acceptable limits for the trigger to be considered active. Parameters t_0 and t_1 define the monitoring start and end times.

2.11.5 Impact of Triggers on System State

Trigger activation affects the overall system state S, indicating a need for corrective actions. The overall system state with active triggers is defined as:

$$S = H(T) = H(t_1, t_2, \ldots, t_k) \tag{2.22}$$

where H is an aggregating function that evaluates system status based on active triggers. The greater the number of active triggers, the higher the probability that the system requires attention and adjustment.

2.11.6 Conflicts between Triggers and Balance Conditions

Certain triggers may conflict with each other. Let $R \subseteq T \times T$ be the set of conflicting trigger pairs. Each pair (t_i, t_j) belongs to R if their simultaneous activation creates interpretation conflicts in assessing system state. Balance conditions are defined to resolve these conflicts:

$$\forall (t_i, t_j) \in R \Rightarrow (\exists c_m : t_i(c_m) = t_j(c_m) = 0) \tag{2.23}$$

where c_m is the condition under which both triggers are inactive, avoiding scenarios where conflicting signals lead to ambiguity in system assessment.

2.11.7 Conclusion

The formalized set of triggers T, including threshold values, temporal parameters, and membership functions, forms the foundation of the anomaly monitoring and detection system. This structure allows identification of metric deviations, maintains system stability, and enables prompt response to critical changes.

2.12 Formalization of the Data Source Set

Let D be the set of data sources that supply the system with essential metrics and information for monitoring and analysis. Data sources include both external monitoring systems and internal databases, enabling real-time collection and processing of system status information. The definition, classification, and impact of data sources on system parameters are formalized as follows.

2.12.1 Definition of the Data Source Set

Define the set of data sources D as the collection of all external and internal systems providing access to data for system state analysis:

$$D = \{d_1, d_2, \ldots, d_n\} \tag{2.24}$$

where d_i represents an individual data source, providing one or more types of system metric data, such as video quality, network traffic, response time, etc. Each data source d_i is defined as a function of available data:

$$d_i = f_i(X, Y) \tag{2.25}$$

where X denotes the set of external data provided by monitoring systems, and Y represents internal data from databases.

2.12.2 Classification of Data Sources by Type and Origin

Data sources are classified by origin into external and internal sources, simplifying their processing and determining their significance to the system:

$$C(D) = \{D_{\text{external}}, D_{\text{internal}}\} \tag{2.26}$$

where D_{external} refers to external sources such as third-party monitoring providers (e.g., Prometheus, Grafana), and D_{internal} includes internal sources, such as company databases or system log files.

2.12.3 Membership Function for Data Source Reliability

Each data source $d_i \in D$ is assigned a membership function $\mu(d_i, C)$, indicating the reliability and importance of that source to the system:

$$\mu(d_i, C) = \frac{1}{1 + e^{-\alpha(x-\beta)}} \tag{2.27}$$

where α and β are parameters of the membership function, depending on the data source type and its relevance to the system. The membership function value indicates the confidence level in the data obtained from the source.

2.12.4 Data Update Interval Parameters

Each data source d_i is assigned temporal parameters for data update frequency, allowing control over the rate at which new information is received:

$$\Delta t_i = \int_{t_0}^{t_1} g(d_i)\, dt \tag{2.28}$$

where Δt_i represents the data update interval for source d_i, and $g(d_i)$ is a function that specifies the data update frequency for the source, based on its characteristics and technical capabilities.

2.12.5 Impact of Data Sources on Overall System State

The system state S depends on the combination of all data sources that determine its key operational parameters. Let S be a function of all data sources D, then the overall system state can be defined as:

$$S = H(D) = H(d_1, d_2, \ldots, d_n) \tag{2.29}$$

where H is an aggregating function that considers the weight and data value from each source d_i. H could be a weighted sum or another aggregating function that reflects the influence of the data sources on the system's state.

2.12.6 Conflicts between Data Sources and Consistency Conditions

Some data sources may provide conflicting information, leading to uncertainty in assessing system status. Let $R \subseteq D \times D$ be the set of pairs of conflicting data sources, where each pair (d_i, d_j) is in R if the data provided by d_i and d_j conflict under the same conditions:

$$\exists c_k \in C : f(d_i, c_k) \neq f(d_j, c_k) \tag{2.30}$$

To resolve such conflicts, consistency conditions are introduced, bringing data from sources d_i and d_j into alignment:

$$\forall (d_i, d_j) \in R \Rightarrow (\exists c_m : f(d_i, c_m) = f(d_j, c_m)) \tag{2.31}$$

where c_m is the condition under which the conflict between sources is resolved, ensuring data consistency and the reliability of the information entering the system.

2.12.7 Conclusion

The formalized set of data sources D, including classification, membership functions, temporal parameters, and consistency conditions, provides a foundation for reliable data collection and system status analysis. This structure accounts for different data types and ensures a high degree of accuracy and consistency in monitoring and managing the system.

2.13 Formalization of the Set of Filtering Rules to Minimize False Positives

Let F be the set of filtering rules applied to minimize the number of false positives in the anomaly detection system. These rules set threshold values and conditions under which metric deviations are not considered anomalies, improving model accuracy and reducing false positives.

2.13.1 Definition of the Set of Filtering Rules

Define the set of filtering rules F as a collection of conditions to minimize false positives:

$$F = \{f_1, f_2, \ldots, f_m\} \tag{2.32}$$

where f_i is an individual filtering rule applying conditions to metrics to exclude minor deviations. Each rule f_i can be represented as a logical condition:

$$f_i(m) = \begin{cases} 1, & \text{if } m \in \text{anomaly and } m \notin \text{filtered} \\ 0, & \text{otherwise} \end{cases} \tag{2.33}$$

where m is a metric value, and filtered is a condition excluding the metric from anomalies if rule f_i is satisfied.

2.13.2 Classification of Filtering Rules by Type and Thresholds

Filtering rules can be classified by their type and impact on anomaly filtering:

$$C(F) = \{F_{\text{threshold}}, F_{\text{trend}}, F_{\text{contextual}}\} \tag{2.34}$$

where $F_{\text{threshold}}$ includes threshold-based rules (e.g., standard deviation from the mean), F_{trend} considers trend changes, and $F_{\text{contextual}}$ accounts for contextual factors such as seasonal or temporal changes in data.

2.13.3 Membership Function for Filtering Reliability Evaluation

Each filtering rule $f_i \in F$ is assigned a membership function $\mu(f_i, C)$, which determines the reliability probability of the rule within the current conditions:

$$\mu(f_i, C) = \frac{1}{1 + e^{-\alpha(x-\beta)}} \tag{2.35}$$

where α and β are parameters of the membership function, defined by the rule type and significance in minimizing false positives. The membership function value indicates the filtering rule's effectiveness.

2.13.4 Temporal Activation Parameters for Filtering Rules

Filtering rules can include temporal parameters to consider the duration of metric deviations before they are considered anomalies. Let $\tau(f_i)$ be the temporal parameter for rule f_i:

$$\tau(f_i) = \int_{t_0}^{t_1} g(m)\, dt \tag{2.36}$$

where $g(m)$ represents the deviation of metric m over time. Parameters t_0 and t_1 define the time interval over which the metric deviation must persist for the filtering rule to activate.

2.13.5 Impact of Filtering Rules on Overall System State

The overall system state S depends on false positive filtering, as it affects anomaly detection accuracy and reliability. Let S be a function of applied filtering rules F, then the system state can be expressed as:

$$S = H(F) = H(f_1, f_2, \ldots, f_m) \tag{2.37}$$

where H is an aggregating function accounting for the influence of each filtering rule on system state assessment. This function may be, for instance, a weighted sum, defining the proportion of false positives excluded through filtering rules.

2.13.6 Conflicts between Filtering Rules and Consistency Conditions

Some filtering rules may conflict, especially if they focus on different data aspects. Let $R \subseteq F \times F$ be the set of conflicting filtering rule pairs, where each element (f_i, f_j) is in R if simultaneously active rules f_i and f_j produce different evaluations:

$$\exists c_k \in C : f_i(m, c_k) \neq f_j(m, c_k) \tag{2.38}$$

To resolve these conflicts, consistency conditions are introduced, aligning the filtering rules:

$$\forall (f_i, f_j) \in R \Rightarrow (\exists c_m : f_i(m, c_m) = f_j(m, c_m) = 0) \tag{2.39}$$

where c_m is the condition resolving the conflict between filtering rules, ensuring coherent and consistent rule application.

2.13.7 Conclusion

The formalized set of filtering rules F, including classification, temporal parameters, and consistency conditions, establishes the foundation for a reliable false positive minimization system. This structure ensures high detection accuracy, allowing prompt responses to real anomalies while avoiding errors due to minor deviations.

2.14 Formalization of the Set of Dynamic Thresholds

Let $M_{\text{dynamic\_threshold}}$ represent the set of dynamic thresholds that define adaptive boundaries for anomaly detection based on historical data and the current system context. These thresholds account for changes in the data over time, enabling more flexible and precise monitoring configurations.

2.14.1 Definition of the Set of Dynamic Thresholds

Define the set of dynamic thresholds $M_{\text{dynamic\_threshold}}$ as a collection of thresholds T_i, each calculated based on historical data and context:

$$M_{\text{dynamic\_threshold}} = \{T_i \mid T_i = f(\text{historical\_data}, \text{context})\} \tag{2.40}$$

where T_i is an individual dynamic threshold, determined by function f that depends on the system's historical data (historical_data) and the current state context (context). This function may consider statistical measures such as mean, median, and standard deviation, or employ more complex predictive algorithms based on time series.

2.14.2 Contextual Adaptation of Dynamic Thresholds

Each threshold T_i adapts to the current context, which includes temporal, seasonal, and functional factors. Let $C(T_i)$ represent the contextual parameters of the threshold, then the dynamic threshold T_i can be defined as:

$$T_i = f(\text{historical\_data}, C(T_i)) \tag{2.41}$$

where $C(T_i)$ is the set of factors influencing the threshold T_i, including temporal parameters (e.g., day of the week, time of day), seasonal variations, and specific system operation conditions.

2.14.3 Adaptive Threshold Function

The function f that defines the threshold T_i can be represented as an adaptive threshold, taking into account past metric values. Let h represent historical data including the time series of metric m, then the function f may be defined as follows:

$$f(h, C(T_i)) = \mu(h) + k \cdot \sigma(h) \tag{2.42}$$

where $\mu(h)$ is the mean of historical data h, $\sigma(h)$ is the standard deviation, and k is a scaling factor, depending on the metric's criticality and its variability within the context $C(T_i)$.

2.14.4 Temporal Parameters for Dynamic Threshold Updates

To ensure relevance, dynamic thresholds T_i are updated at specified intervals. Let Δt_i represent the update interval for threshold T_i, then dynamic threshold updating can be expressed as:

$$\Delta t_i = \int_{t_0}^{t_1} g(T_i, \text{new\_data})\, dt \tag{2.43}$$

where $g(T_i, \text{new\_data})$ is a function describing the frequency of threshold T_i updates based on new incoming data and detection accuracy requirements.

2.14.5 Impact of Dynamic Thresholds on Anomaly Detection

Dynamic thresholds T_i directly influence the anomaly detection process as they adapt to changing conditions. The overall system state S, considering the influence of dynamic thresholds, can be expressed through a state function:

$$S = H(T) = H(T_1, T_2, \ldots, T_n) \tag{2.44}$$

where H is an aggregating function that determines the system state based on current dynamic threshold values. H can account for the number and degree of deviations detected based on thresholds T_i, allowing the system to adapt to changing operating conditions and reduce false positives.

2.14.6 Conflicts between Dynamic Thresholds and Consistency Conditions

Dynamic thresholds can sometimes conflict if their values contradict one another under identical conditions. Let $R \subseteq M_{\text{dynamic\_threshold}} \times M_{\text{dynamic\_threshold}}$ be the set of conflicting threshold pairs, where each element (T_i, T_j) is in R if the values T_i and T_j conflict under identical conditions:

$$\exists c_k \in C : f(T_i, c_k) \neq f(T_j, c_k) \tag{2.45}$$

To resolve conflicts between thresholds, consistency conditions are introduced, where threshold values are aligned:

$$\forall (T_i, T_j) \in R \Rightarrow (\exists c_m : f(T_i, c_m) = f(T_j, c_m)) \tag{2.46}$$

where c_m is a specific condition under which the conflict between thresholds is resolved, ensuring consistent and reliable monitoring system operation.

2.14.7 Conclusion

The formalized set of dynamic thresholds $M_{\text{dynamic\_threshold}}$, including functions for adaptation to historical data, context, and consistency conditions, establishes a foundation for a flexible and reliable anomaly detection system. This structure allows the system to minimize false positives, adapt to changing conditions, and achieve high accuracy in anomaly detection.

2.15 Formalization of the Time Series Prediction Function

Let $P_{\text{prediction}}$ be a predictive function based on the system's time series, enabling the estimation of future metric values and the detection of potential anomalies before they occur. This predictive function uses historical data and trends to build a forecasting model that adapts to changing conditions.

2.15.1 Definition of the Predictive Function

Define the predictive function $P_{\text{prediction}}$ as the result of applying function g to the set of time series $M_{\text{time\_series}}$:

$$P_{\text{prediction}} = g(M_{\text{time\_series}}) \tag{2.47}$$

where $M_{\text{time\_series}} = \{m_1(t), m_2(t), \ldots, m_n(t)\}$ represents the set of time series of system metrics $m_i(t)$, depending on time t. Each time series $m_i(t)$ describes the metric m_i value changes over time, allowing assessment of current trends and probable future values.

2.15.2 Time Series-Based Forecasting Function

The forecasting function g is based on time series analysis and may use methods such as moving average, exponential smoothing, or autoregressive models (e.g., ARIMA). Let us formalize the forecasting function for time series $m_i(t)$:

$$g(m_i(t)) = \hat{m}_i(t + \Delta t) = a_0 + \sum_{k=1}^{p} a_k m_i(t - k) + \epsilon \tag{2.48}$$

where $\hat{m}_i(t + \Delta t)$ is the predicted metric m_i value at time $t + \Delta t$, a_k are model coefficients, p is the model order, and ϵ is random error or noise, assumed to be white noise with mean zero and variance σ^2.

2.15.3 Adaptation of the Predictive Model to Changing Conditions

To account for current changes in the data, the predictive model is periodically updated based on new data. Let θ represent the model parameters, then the parameter update can be expressed as:

$$\theta_{t+1} = \theta_t + \alpha \cdot \nabla_\theta L(\theta, M_{\text{time\_series}}) \tag{2.49}$$

where α is the learning rate, and $L(\theta, M_{\text{time\_series}})$ is the loss function that minimizes the difference between predicted and actual time series values, ensuring model adaptation to changes in metric time series.

2.15.4 Temporal Parameters for Prediction Updates

Predictions are recalculated at specified intervals to ensure data relevance. Let Δt_{update} be the prediction update interval for the function $P_{\text{prediction}}$, then the prediction recalculation can be represented as:

$$P_{\text{prediction}}(t + \Delta t_{\text{update}}) = g(M_{\text{time\_series}}(t + \Delta t_{\text{update}})) \tag{2.50}$$

where Δt_{update} is determined by data characteristics and time series change frequency.

2.15.5 Impact of Forecasted Values on Overall System State

The forecasted values $\hat{M}_{\text{time\_series}}$ obtained from $P_{\text{prediction}}$ affect the system state S, allowing early detection of potential anomalies. Let S be a function depending on forecasted values, then the system state can be expressed as:

$$S = H(\hat{M}_{\text{time\_series}}) \tag{2.51}$$

where H is an aggregating function that considers both current metric values and forecasted data. This allows the system to assess potential risks and take preventive measures.

2.15.6 Conflicts between Forecasted and Current Metric Values

Significant deviations between forecasted and current metric values may indicate a conflict, signaling a potential anomaly. Let $R \subseteq M_{\text{time\_series}} \times \hat{M}_{\text{time\_series}}$ be the set of conflicting pairs, where each pair $(m_i, \hat{m}_i)$ satisfies:

$$\exists c_k \in C : |m_i(t) - \hat{m}_i(t)| > \delta \tag{2.52}$$

where δ is the threshold for permissible deviation. To resolve conflicts, corrective actions are introduced, aiming to align current and forecasted values or signal a potential anomaly.

2.15.7 Conclusion

The formalized predictive function $P_{\text{prediction}}$, based on time series, allows the system to forecast the future state of metrics, considering historical data and trends. Model adaptation and consideration of conflicts between forecasted and current values enable the system to minimize risks and enhance anomaly detection reliability.

2.16 Formalization of the System Metric Hierarchy

Let $H_{\text{metric\_hierarchy}}$ represent a hierarchical structure of metrics, grouping metrics by their level of criticality and significance to the system. This hierarchy allows for prioritizing metrics, enabling more precise management and monitoring of system status. The hierarchy divides metrics into critical, warning, and informational levels, formalized as follows.

2.16.1 Definition of the Metric Hierarchy

Define the hierarchical structure of metrics $H_{\text{metric\_hierarchy}}$ as a grouping of metrics by criticality level:

$$H_{\text{metric\_hierarchy}} = \text{group}(\{M_{\text{critical}}, M_{\text{warning}}, M_{\text{info}}\}) \tag{2.53}$$

where M_{critical}, M_{warning}, and M_{info} are sets of metrics at the critical, warning, and informational levels, respectively. These metric sets have varying characteristics and different impacts on the system's state.

2.16.2 Critical Metrics M_{critical}

The set M_{critical} includes metrics whose deviations indicate serious system issues requiring immediate attention. For each metric $m_i \in M_{\text{critical}}$, its impact on system state is defined as a function of its current value:

$$I(m_i) = w_i \cdot m_i \tag{2.54}$$

where w_i is a weighting coefficient reflecting the importance of metric m_i within the system. If $I(m_i)$ exceeds a permissible threshold T_{critical}, an emergency alert is triggered:

$$\text{If } I(m_i) > T_{\text{critical}}, \text{ then a critical alert is activated.} \tag{2.55}$$

2.16.3 Warning Metrics M_{warning}

The set M_{warning} includes metrics whose deviations may indicate potential risks requiring monitoring, but not immediate action. Each metric $m_j \in M_{\text{warning}}$ is evaluated considering its weight and the warning threshold T_{warning}:

$$I(m_j) = w_j \cdot m_j \tag{2.56}$$

If $I(m_j)$ exceeds the threshold T_{warning}, a warning is generated:

$$\text{If } I(m_j) > T_{\text{warning}}, \text{ then a warning of potential risks is activated.} \tag{2.57}$$

2.16.4 Informational Metrics M_{info}

The set M_{info} includes metrics that provide information about the overall system state without significantly impacting core processes. Values of metrics $m_k \in M_{\text{info}}$ are monitored but do not trigger warnings or alerts. These metrics can be described as:

$$I(m_k) = w_k \cdot m_k \tag{2.58}$$

where w_k is a weight reflecting the significance of informational metric m_k in the overall system assessment.

2.16.5 Aggregated Impact of Metrics on System State

The system state S is formed based on the aggregated value of all metrics within $H_{\text{metric\_hierarchy}}$. Let $H_{\text{metric\_hierarchy}}(S)$ be the function that defines the overall system state, then the aggregated metric impact is represented as:

$$S = H_{\text{metric\_hierarchy}}(M_{\text{critical}}, M_{\text{warning}}, M_{\text{info}}) \tag{2.59}$$

The function $H_{\text{metric\_hierarchy}}$ may be a weighted sum or another aggregating function that considers the varying criticality levels of metrics, allowing the system to respond appropriately to state changes.

2.16.6 Conflicts between Metrics of Different Levels

Conflicts may arise when metrics from different hierarchy levels create contradictory indications, complicating the correct interpretation of the system state. Let $R \subseteq H_{\text{metric\_hierarchy}} \times H_{\text{metric\_hierarchy}}$ be the set of conflicting metrics, where each conflict (m_i, m_j) occurs if values of metrics $m_i \in M_{\text{critical}}$ and $m_j \in M_{\text{warning}}$ contradict under the same conditions:

$$\exists c_k \in C : I(m_i) \neq I(m_j) \tag{2.60}$$

To resolve conflicts, consistency conditions are introduced, prioritizing more critical metrics:

$$\forall (m_i, m_j) \in R \Rightarrow I(m_i) \text{ takes precedence over } I(m_j), \text{ if } m_i \in M_{\text{critical}} \tag{2.61}$$

These conditions ensure metric hierarchy consistency and prevent misinterpretations of system state.

2.16.7 Conclusion

The formalized metric hierarchy $H_{\text{metric\_hierarchy}}$, which includes critical, warning, and informational metrics, establishes a foundation for effective monitoring and system management. This structure allows the system to prioritize metrics based on criticality and respond promptly to state changes.

2.17 Formalization of the Metric Correlation Analysis Function

Let $C_{\text{correlation\_analysis}}$ be a correlation analysis function designed to detect relationships between two system metrics M_i and M_j. Correlation analysis helps assess the influence of one metric on another, valuable for understanding system behavior and identifying potential anomalies caused by correlated metric deviations.

2.17.1 Definition of the Correlation Analysis Function

Define the correlation analysis function $C_{\text{correlation\_analysis}}$ as function h, which calculates the correlation coefficient between metrics M_i and M_j:

$$C_{\text{correlation\_analysis}} = h(M_i, M_j) \tag{2.62}$$

where M_i and M_j are time series of system metric values represented as functions of time t: $M_i = \{m_{i,1}, m_{i,2}, \ldots, m_{i,n}\}$ and $M_j = \{m_{j,1}, m_{j,2}, \ldots, m_{j,n}\}$.

2.17.2 Pearson Correlation Coefficient

For measuring linear relationships between metrics M_i and M_j, the Pearson correlation coefficient r is applied:

$$r_{ij} = \frac{\sum_{k=1}^{n}(m_{i,k} - \bar{M}_i)(m_{j,k} - \bar{M}_j)}{\sqrt{\sum_{k=1}^{n}(m_{i,k} - \bar{M}_i)^2} \cdot \sqrt{\sum_{k=1}^{n}(m_{j,k} - \bar{M}_j)^2}} \tag{2.63}$$

where $\bar{M}_i$ and $\bar{M}_j$ are the mean values of metrics M_i and M_j respectively, and n is the number of observations. The value of r_{ij} ranges from -1 to 1, where $r_{ij} = 1$ indicates full positive correlation, $r_{ij} = -1$ full negative correlation, and $r_{ij} = 0$ no linear relationship.

2.17.3 Spearman Rank Correlation Coefficient

For nonlinear relationships, the Spearman rank correlation coefficient ρ, based on value ranking, can be used:

$$\rho_{ij} = 1 - \frac{6 \sum_{k=1}^{n} (R(m_{i,k}) - R(m_{j,k}))^2}{n(n^2 - 1)} \tag{2.64}$$

where $R(m_{i,k})$ and $R(m_{j,k})$ are ranks of metric values $m_{i,k}$ and $m_{j,k}$. Spearman's coefficient, like Pearson's, ranges from -1 to 1 and is more robust to outliers, suitable for nonlinear dependencies.

2.17.4 Correlation Significance Conditions

To verify the significance of the correlation coefficient r_{ij}, a t-test is applied with test statistic t:

$$t = \frac{r_{ij}\sqrt{n-2}}{\sqrt{1 - r_{ij}^2}} \tag{2.65}$$

where n is the number of observations. The resulting t value is compared to the critical value $t_{\alpha/2,n-2}$, where α is the significance level. If $|t| > t_{\alpha/2,n-2}$, the correlation is considered statistically significant at level α.

2.17.5 Impact of Correlation Analysis on System State

The results of correlation analysis between metrics M_i and M_j affect the overall system state S, as high correlation values may indicate causal or interdependent relationships between metrics. Let S be a function dependent on correlation analysis results; then, the system state can be expressed as:

$$S = F(C_{\text{correlation\_analysis}}(M_i, M_j)) \tag{2.66}$$

where F is a function that adjusts the system state assessment based on correlation analysis results, identifying potential interrelated anomalies.

2.17.6 Conflicts between Correlated Metrics

In cases of strong correlation between metrics M_i and M_j, conflicts may arise in interpreting system state, especially if both metrics exhibit deviations. Let $R \subseteq C_{\text{correlation\_analysis}} \times C_{\text{correlation\_analysis}}$ represent the set of correlated metrics, satisfying:

$$\exists c_k \in C : |r_{ij}| > \gamma \tag{2.67}$$

where γ is the correlation threshold indicating strong correlation. If this threshold is exceeded, the system should account for the correlation in state assessment to prevent double-counting anomalies.

2.17.7 Conclusion

The formalized correlation analysis function $C_{\text{correlation\_analysis}}$, based on Pearson and Spearman coefficients, enables the system to detect interdependencies between metrics. By considering correlated metrics' impact on system state assessment, this approach aids in identifying the root causes of anomalies and preventing false positives.

Bibliography

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006. A comprehensive resource covering statistical techniques in machine learning, including methods applicable to anomaly detection, such as Bayesian inference and support vector machines.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009. An extensive survey on anomaly detection methodologies, examining approaches across various application domains and discussing different types of anomalies and detection algorithms.

[3] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2018. An accessible text on time series forecasting techniques, with practical insights into predictive modeling and adaptive thresholding strategies for anomaly detection.

[4] Y. Zhu, D. Guo, H. Li, et al., "Anomaly Detection in Data Streams Using Dynamic Thresholds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1283–1296, 2018. This paper discusses the application of dynamic thresholding for real-time data streams, outlining methods to set context-aware, adaptive thresholds.

[5] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000. Introduces the Local Outlier Factor (LOF) method, a density-based approach useful for anomaly detection in multidimensional datasets by assessing local deviations.

[6] T. Dunning and E. Friedman, *Practical Machine Learning: A New Look at Anomaly Detection*. O'Reilly Media, 2014. A practical guide on machine learning for anomaly detection, including the use of clustering, outlier analysis, and real-time monitoring techniques.

[7] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014. This survey covers methods for adapting to concept drift in data streams,

a significant challenge in maintaining accuracy in anomaly detection models over time.

[8] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*. D. Van Nostrand Company, 1931. A foundational text on quality control and statistical process control, introducing principles of thresholding and early detection in process monitoring.