Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

Proactive Monitoring of IT System's Critical Parameter

G.Priyadharshini[1], Dr. K. Shyamala[2]

[1]Research Scholar, PG & Research Department of Computer Science, Dr. Ambedkar Government Arts College, Chennai, India
sushpriya@yahoo.com
[2]Associate Professor, PG & Research Department of Computer Science, Dr. Ambedkar Government Arts College, Chennai, India
shyamalakannan2000@gmail.com

Abstract: The new IT structures demand active monitoring systems that have the ability to identify abnormalities in important parameters of the system, before the failures affect the performance and the reliability. This paper introduces a new framework called the UACP-PM (Uncertainty-Aware Contrastive Prototype-based Proactive Monitoring) to monitor the IT system metrics intelligently through time-series learning and adaptive decision mechanisms. The suggested framework starts with preprocessing multi-source telemetry data by filtering noise, eliminating outliers, imputing missing values, and normalizing. A time-series building method based on a sliding window is used to learn time-based dependencies, after which a Temporal Convolutional Network with a dual attention mechanism is used to extract features. The TS2Vec (Time Series to Vector) contrastive representation learning strengthens the features, and a prototype-guided transformer classifier is used to identify fault states. Evidential deep learning is built in to measure uncertainty of prediction and enhance reliability of alerts. Lastly, real-time monitoring and constant adjustment is made possible by visualization dashboards. Experimental assessment shows that the suggested framework has an anomaly detection rate of 98.2% with a precision of 98.1, recall of 98.0 and an F1-score of 98.2, and an AUC value of 99.9, thus minimizing false alarms and enabling the sound proactive decision-making in dynamic IT settings.

Keywords: Proactive monitoring, anomaly detection, IT system monitoring, time-series analysis, machine learning, uncertainty-aware prediction, predictive maintenance, real-time alerts.

1. INTRODUCTION

The growing complexity of contemporary IT infrastructures such as cloud computing platforms, data centers, and IoT-enabled systems has seen the need to ensure system reliability and operational continuity become a significant challenge. Unforeseen failures or abnormal behavior of such an environment can cause severe service outage, loss in revenues, and a decrease in user satisfaction. Conventional reactive monitoring methods identify system failures after they happen, which cannot be tolerated in high-stakes and large-scale IT systems. In a bid to overcome such shortcomings, proactive monitoring of important parameters of the system has become one of the most important strategies of anticipating possible failures and allowing timely intervention. Successful proactive monitoring depends on the capacity to analyze huge volumes of high-dimensional telemetry data emitted by numerous components of the system. The deep learning models based on ensembles have been found to be effective in capturing minute patterns in these measures enabling the early detection of anomalies that can occur before system failures. On the same note, predictive models created with large-scale IT environments can be used to identify anomalies within the system performance before they become critical enough to warrant preventive actions and resource optimization on the part of the operators.

Data accuracy of proactive monitoring frameworks is heavily reliant on quality and consistency of telemetry data. Noise, missing values and outliers are common in raw system metrics, and they can negatively impact the performance of the model unless they are treated with proper methods. High-end preprocessing methods like noise filtering, outlier elimination and normalization are thus necessary to generate the best inputs to predictive models. These methods allow models to detect both short-term and long-term patterns by transforming raw metrics into

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

structured sequences of time-series, enhancing detection sensitivity and minimizing false positives. Anomaly prediction in real-time will enable the operators to predict possible problems before they arise, by keeping the critically operating parameters within safe operating limits. The frameworks based on AI have an opportunity to dynamically rank anomalies based on their severity and the effect on the operations, which can be used to effectively make decisions and reduce human involvement.

Uncertainty of predictive modeling is another important factor to consider. Uncertainty-sensitive approaches, like evidential deep learning, offer probabilistic evaluations of system states, differentiating between confident and uncertain forecasts.

1.1 Main Contribution

The primary contributions of this work are related to the creation of an intelligent and uncertainty-aware proactive monitoring of critical parameters of IT systems with the help of sophisticated time-series learning and adaptive anomaly detection methods. The major contributions can be outlined as follows:

- UACP-PM model is introduced that combines time-series modeling and Temporal Convolutional Network, as well as dual attention, to learn short-term fluctuations and long-term behavioral patterns of IT system metrics.
- To enhance the accuracy of anomaly detection results and minimize the reliance on labeled fault data, a contrastive representation learning approach based on TS2Vec with the help of a prototype-guided transformer classifier is proposed.
- An uncertainty-aware proactive alert generation system with Evidential Deep Learning and real-time visualization and continuous learning support is created to be more reliable, less prone to false alarms, and preserve the monitoring performance in changing system conditions.

2. LITERATURE REVIEW

Recent years have seen a great interest in the detection and prediction of anomalies in IT and cyber-physical systems, especially concerning cloud infrastructures, IoT environments, and complex dynamical systems. Somma et al. [1] suggested an organized theory of anomaly detection in intricate dynamical systems based on embedding theory and physics-inspired consistency.. Ugolini et al. [2] created a explainable probabilistic anomaly detection system to monitor the condition of helicopter transmissions. The framework ensures that the operators trust it and make informed decisions by offering probabilistic confidence measures on the detected abnormalities. Chourasiya et al. [3] introduced a developed system log analyzer based on the LSTM and transformer networks to detect anomalies and conduct cyber-forensic investigations. Their method is the best in the processing of sequential log data to identify performance and security anomalies in the cloud systems. Ortiz-Garcés et al. [4] introduced a self-organizing cyber-physical security middleware in IoT systems, which is a hybrid of anomaly identification and adaptive response elements. Xin et al. [5] addressed robust performance anomaly detection in the cloud environment by proposing an ensemble learning-based model to predict performance degradation of cloud applications.

He et al. [6] introduced a parallel framework using graph transformer dynamics to localize anomalies of a system in a cloud infrastructure. Jiang [7] proposed an artificial intelligence (AI)-based adaptive anomaly detection system in IaaS cloud virtual machines. Lastly, Gu et al. [8] created KPIRoot+ which is a framework of detecting anomalies and analyzing root causes in a large-scale cloud environment.

In general, these papers reflect the evolution of the classical methods of anomaly detection to intelligent, adaptive, and explainable models that can process the data of large and multi-source IT systems

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

3. PROPOSED UNCERTAINTY-AWARE TIME-SERIES BASED PROACTIVE MONITORING FRAMEWORK FOR CRITICAL IT SYSTEM PARAMETERS

The developed Uncertainty-Aware Contrastive Prototype-Based Proactive Monitoring (UACP-PM) framework aims to facilitate robust and prompt identification of anomalies in vital parameters of IT systems with the help of a planned multi-step processing pipeline. First, the data of system telemetry is taken out of the Network Anomaly Detection Dataset that includes a variety of performance-related parameters that define both normal and abnormal operations. The resulting data are then subjected to a preprocessing phase which involves noise, outlier, missing value, and min-max normalization to enhance the quality of data and maintain uniformity in the monitoring variables which are heterogeneous. The cleaned data are then converted into the structured temporal sequences after preprocessing with the help of a sliding window method and multi-resolution segmentation, enabling the framework to identify the short-term fluctuations and long-term patterns of behavior in the system performance.

This is followed by high-level temporal features with a Temporal Convolutional Network (TCN) with a dual attention mechanism comprising of temporal attention and feature attention that further improves the capacity of the system to emphasize the most significant system behaviors in detection of anomalies. To further enhance the strength of representation and to minimize the reliance on labeled fault samples, contrastive representation learning with TS2Vec is utilized to produce discriminative representation of normal and abnormal system states. Such refined embeddings are then fed into a hybrid classification approach that learns a prototype guided energy-based classifier alongside a transformer-based classifier, enabling the fault-prone system conditions to be accurately recognized as a result of the contextual dependencies among monitoring parameters.

The framework uses Evidential Deep Learning (EDL) to predict anomalies with uncertainty, to decrease false alarms and increase trust in proactive alerts, by predicting belief, disbelief and uncertainty in the prediction of anomalies and to improve the reliability of predictions. Fig. 1 depicts the architecture process of the suggested UACP-PM framework of uncertainty-driven proactive anomaly monitoring in dynamic IT environments.

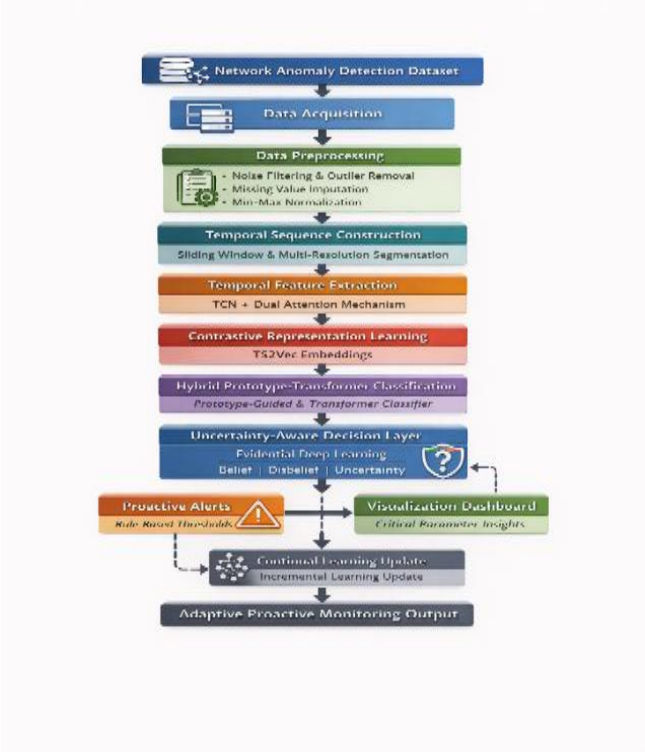


Fig. 1. Workflow diagram of the proposed UACP-PM framework for uncertainty-aware proactive anomaly detection and adaptive monitoring in dynamic IT system environments

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

3.1 Dataset Collection and Description

To create a useful proactive monitoring system to detect anomaly in key parameters in the IT system, a relevant and representative data set is needed to represent a variety of operational attributes of network-level telemetry signals. The primary data source to be used in the proposed UACP-PM framework training and testing is the Network Anomaly Detection Dataset that was obtained at the Network Anomaly Detection Dataset. This dataset includes various system-level monitoring characteristics of normal and anomalous behavior patterns seen in a network infrastructure setting; it is appropriate in proactive anomaly prediction activities.

Let the collected monitoring dataset be represented as in eq. (1), Where, $X = \{x_1, x_2, x_3, \dots, x_N\}$ denotes the set of multivariate monitoring samples and $Y = \{y_1, y_2, y_3, \dots, y_N\}$ refers to corresponding system condition labels indicating normal and anomalous operational states. Every monitoring instance x_i is composed of several parameters of performance characteristics like throughput variation, packet-level statistics, traffic behavior parameters and system activity parameters as a result of time. All these parameters define the health of the IT infrastructure being monitored.

$$\mathcal{D} = \{X, Y\} \quad (1)$$

More specifically, each monitoring sample can be expressed as a feature vector in eq. (2), where d refers to number of monitored critical system parameters and N refers to total number of observations available in the dataset. These multivariate monitoring capabilities record changes in system behavior over time scales and are used to provide the basis of downstream time-series modeling and anomaly prediction.

$$x_i = [x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, \dots, x_i^{(d)}] \in \mathbb{R}^d \quad (2)$$

As proactive monitoring involves learning temporal dependencies between changing system parameters, the dataset is organized into serial monitoring observations in discrete time steps. Based on this, the dataset will be expressed in time form as in the equation (3), where T refers to total number of observation time steps and each temporal observation is defined as in eq. (4) representing the system monitoring matrix at time step t.

$$X = \{X^{(1)}, X^{(2)}, X^{(3)}, \dots, X^{(T)}\} \quad (3)$$

$$X^{(t)} \in \mathbb{R}^{N \times d} \quad (4)$$

This temporal structure allows the structure to represent changes in behavioral transitions between normal and abnormal operational states.

3.2 Data Preprocessing

Once the datasets are acquired, the telemetry information obtained has to be systematically preprocessed to guarantee reliability, consistency, and compatibility with the temporal learning models. Because the raw monitoring data is usually noisy in nature, there may be missing data, outliers, and scale variations across heterogeneous system parameters, a powerful preprocessing pipeline is required to enhance the downstream performance of anomaly detection. Noise filtering, outlier removal, missing value imputation, and minmax normalization as part of preprocessing in the proposed UACP-PM framework convert raw monitoring signals into structured and learning-ready forms. Assume that the multivariate monitoring data collected can be represented as in eq. (5), where N refers to number of monitoring samples and d refers to number of critical system parameters. Preprocessing is aimed at getting a fine dataset as in eq. (6) where noise, inconsistencies, and scale differences are reduced and then temporal modeling is done.

$$X = \{x_1, x_2, x_3, \dots, x_N\}, x_i \in \mathbb{R}^d \quad (5)$$

$$X^* = \{x_1^*, x_2^*, x_3^*, \dots, x_N^*\} \quad (6)$$

The first step is noise filtering, to reduce high-frequency variations that can occur because of measurement variation or transmission distortion in telemetry signals. Let the original monitoring signal be denoted as $x(t)$. A

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

filtering operation that reduces noise is a smoothing-based operation that is performed to come up with a noise-reduced signal, which is represented in eq. (7), where k refers to smoothing window size. The filtering operation enhances the stability of the signal, still maintaining the valuable temporal features that are needed to identify anomalies.

$$x_f(t) = \frac{1}{2k+1} \sum_{i=-k}^{k} x(t+i) \quad (7)$$

After removing noise, outliering is used to remove the abnormal observations that are greatly outliers to the normal behavioral distributions and can adversely affect training of the model. For each monitoring parameter $x^{(j)}$, the statistical deviation is calculated using the standardized score defined as in eq. (8), where $\mu_j$and σ_j refers to mean and standard deviation of the j^{th}monitoring parameter.

$$z_i^{(j)} = \frac{x_i^{(j)} - \mu_j}{\sigma_j} \quad (8)$$

Observations satisfying $| z_i^{(j)} | > \tau$ are considered outliers and are removed or replaced, where τdenotes the predefined threshold value. As the real-world monitoring data are often incomplete, therefore missing value imputation is carried out to ensure continuity of the dataset. Let $x_i^{(j)}$represent a missing observation. It is estimated with statistical mean-based estimation in form of eq. (9), where N_j refers to number of valid observations available for the j^{th}parameter.

$$x_i^{(j)} = \frac{1}{N_j} \sum_{k=1}^{N_j} x_k^{(j)} \quad (9)$$

This is done to make sure that there is no temporal discontinuity between the temporal sequences to be modeled. Finally, to eliminate scale variations among heterogeneous monitoring parameters, min–max normalization is applied. All the monitoring features are converted to a standard range by using eq. (10), where $x_{\min}$and $x_{\max}$ refers to minimum and maximum values of the j^{th}monitoring parameter.

$$x_i^{(j)'} = \frac{x_i^{(j)} - x_{\min}}{x_{\max} - x_{\min}} \quad (10)$$

This normalization phase provides equal scaling of features and enhances convergence of deep learning models in subsequent phases of the proposed framework. The preprocessing phase will result in a clean and standardized monitoring dataset.

3.3 Time-Series Construction Using Sliding Window and Multi-Resolution Segmentation

After the preprocessing phase, the filtered monitoring data are converted into organized time series so that it can be modeled effectively to understand the dynamic behavior of a system in a time-dependent manner. Because the critical IT system parameters change dynamically and have both short-term changes and long-term patterns of operation, the monitoring signals need to be cast to a sequential format that allows learning of the temporal features. Within the proposed UACP-PM framework, this transformation is accomplished with the help of a sliding window methodology with multi-resolution segmentation, which enables the framework to obtain the temporal dependencies at various scales of observation. Arrange the pre-processed monitoring observations available at the last stage in time according to the definition in eq. (11), where each observation $x_t^* \in \mathbb{R}^d$ refers to set of $d$normalized system parameters recorded at time step t.

$$\{x_1^*, x_2^*, x_3^*, \dots, x_N^*\} \quad (11)$$

To construct temporal segments suitable for learning sequential relationships, a sliding window of length w is applied across the monitoring sequence. Based on this, the temporal segment that is produced at time step tis is given by in eq. (12), where $S_t \in \mathbb{R}^{w \times d}$ represents a multivariate time-series segment describing system behavior within the observation window.

$$S_t = \{x_t^*, x_{t+1}^*, \dots, x_{t+w-1}^*\} \quad (12)$$

The movement of the window through the monitoring sequence with stride s produces a set of overlapping time

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

segments that allow the framework to effectively record local changes and sudden changes in behavior. Nevertheless, real-world IT infrastructures tend to have patterns in the monitoring data with different temporal resolutions. Thus, to represent the fine-grained variations and long-term dependency of behavior, multi-resolution segmentation is integrated into the time-series construction process. Instead of using a single observation window, multiple window sizes $w_1, w_2, \dots, w_r$ are considered to represent system behavior at different temporal scales. For each resolution level r, the constructed temporal representation is represented in eq. (13) which enables the learning framework to analyze monitoring signals across short, medium, as well as long-duration observation intervals simultaneously.

$$S_t^{(r)} \in \mathbb{R}^{w_r \times d} \quad (13)$$

The proposed framework, which combines sliding window segmentation and multi-resolution temporal representation, is effective in capturing the short-lived anomalies, a gradual degradation trend, and changing operational traits of the key parameters of the IT system. These sequence timings give the required input data to the next stage Temporal Convolutional Network with dual attention mechanism that conducts hierarchical feature learning to generate proactive anomaly detection at the following phase of the suggested monitoring context.

3.4 Feature Extraction Using Temporal Convolutional Network with Dual Attention Mechanism

Upon the generation of structured temporal sequence with the help of sliding window and multi-resolution segmentation, the following step of the suggested UACP-PM system is the extraction of the discriminative temporal features based on the multivariate monitoring signals. As critical IT system parameters are subject to complex series of sequential dependencies over time, standard feature extraction methods are inadequate in order to exploit long-range temporal dependencies. Thus, a TCN (Temporal Convolutional Network) with a dual attention mechanism comprising of temporal attention and feature attention to learn high level representations on the basis of system monitoring sequences. Assume that the time sequence of the received signal of the last step is denoted by the following expression, where $w$denotes the window length and $d$represents the number of monitored system parameters.

$$S_t \in \mathbb{R}^{w \times d} \quad (14)$$

The goal of the TCN is to learn hierarchical temporal dependencies of these multivariate monitoring sequences without losing cause-effect relationships between observations. The TCN utilizes causal dilated convolutions to ensure that predictions at time step t depend only on current and previous observations. The output of the dilated convolution operation at layer $l$can be expressed as in eq. (15) where $W_i^{(l)}$ refers to convolutional filter weights, $b^{(l)}$ refers to bias term, k refers to kernel size, and d_l refers dilation factor at layer l.

$$h_t^{(l)} = \sum_{i=0}^{k-1} W_i^{(l)} \cdot h_{t-d_l i}^{(l-1)} + b^{(l)} \quad 15)$$

The dilation factor grows exponentially between layers, and allows the network to learn long-range temporal dependence efficiently without growing the computational complexity dramatically. To further increase capacity of feature learning, the temporal representations extracted are optimized with a dual attention mechanism, which enables the model to selectively attend to the most informative time steps and monitoring parameters that aid in anomaly detection. The initial element of the dual attention module is the temporal attention that gives time steps adaptive importance weights within each temporal segment. Denote the intermediate temporal feature representation by in eq. (16)

$$H = \{h_1, h_2, \dots, h_w\} \quad (16)$$

The temporal attention weights are evaluated as in eq. (17) where the attention score $e_t$is defined as in eq. (18).

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{w} \exp(e_i)} \quad (17)$$

$$e_t = v^{\top} \tanh(W_h h_t + b_h) \quad (18)$$

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

Here, W_h, v, and b_h refers to learnable parameters of the temporal attention module. These weights can emphasize the key time intervals that are connected with the abnormal system behavior. The second element is feature attention that gives adaptive significance to various monitoring parameters in each observation window. Assume the feature-level representation can be formulated as in eq. (19) and the weights in the feature attention are computed as in eq. (20)., where W_f, u, and b_f refers to learnable feature-attention parameters.

$$H_f = \{f_1, f_2, \dots, f_d\} \quad (19)$$

$$\beta_j = \frac{\exp(g_j)}{\sum_{i=1}^{d} \exp(g_i)}; \quad g_j = u^{\top} \tanh(W_f f_j + b_f \quad (20)$$

This process allows the framework to give precedence to the most influential system parameters in predicting anomalies. Lastly, the refined feature representation, which has been achieved with the implementation of temporal and feature attention mechanisms, can be represented as in eq. (21), where Z refers to attention-enhanced feature embedding capturing both temporal dependencies and parameter-level importance.

$$Z = \sum_{t=1}^{w} \alpha_t \sum_{j=1}^{d} \beta_j h_{t,j} \quad (21)$$

These learned representations serve as powerful and informative inputs of the next contrastive representation learning with TS2Vec, which further enhances the discrimination between normal and abnormal system state in the next phase of the proposed proactive monitoring framework.

3.5 Representation Refinement Using Contrastive Learning (TS2Vec)

The second step of the proposed UACP-PM framework, based on the extraction of temporal features by the Temporal Convolutional Network with dual attention mechanism, is the refinement of the learned representations with the help of the Time Series to Vector (TS2Vec) contrastive representation learning. This enhances the generalization ability of the framework to different conditions of operation. Denote the attention-enhanced temporal feature representation of the stage above with in eq. (22), where each z_i refers to embedding corresponding to a temporal segment constructed from monitoring observations.

$$Z = \{z_1, z_2, \dots, z_M\} \quad (22)$$

TS2Vec aims to achieve this goal by learning discriminative representations by maximizing similarity between contextually related segments and reducing similarity between unrelated segments in the time-series structure. During contrastive representation learning, augmented time-segments are formed in pairs out of the identical sequence of monitoring. represent two augmented views of a segment in the form of eq. (23).

$$z_i^{(a)} \text{and} z_i^{(b)} \quad (23)$$

These augmented representations are positive pairs but those representations of other temporal segments are negative pairs. The cosine similarity in the pairs is calculated as in the equation (24) to determine the similarity between the pairs.

$$\text{sim}(z_i^{(a)}, z_i^{(b)}) = \frac{z_i^{(a)} \cdot z_i^{(b)}}{\|z_i^{(a)}\| \|z_i^{(b)}\|} \quad (24)$$

The TS2Vec model optimizes the embedding space with a contrastive loss function that pushes similar pairs to be closer and pushes different pairs apart. The objective function can be expressed as in eq. (25), where τ refers to temperature scaling parameter controlling embedding sensitivity and K refers number of contrastive samples considered during training.

$$\mathcal{L}_{contrastive} = -\log \frac{\exp\left(\text{sim}(z_i^{(a)}, z_i^{(b)})/\tau\right)}{\sum_{j=1}^{K} \exp\left(\text{sim}(z_i^{(a)}, z_j^{(b)})/\tau\right)} \quad (25)$$

This contrastive optimization procedure enables the TS2Vec encoder to acquire hierarchical representations of time that retain the local contextual relationships and the global structural features of monitoring sequences. The resulting contrast-enhanced feature representations offer informative and compact embeddings that are used as good inputs in the following prototype-directed transformer-based classification module, which achieves the

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

correct identification of fault-prone operating states in the second step of the proposed proactive monitoring system.

3.6 Fault Classification Using Prototype-Guided Energy-Based Transformer Classifier

Once refined temporal embeddings are produced with the help of TS2Vec-based contrastive representation learning, the second step of the proposed UACP-PM framework is to detect fault-prone conditions of the system with the help of a hybrid classification strategy based on a prototype-guided energy-based classifier and a transformer-based one. This type of hybrid design allows the framework to enhance the accuracy of classification by considering both the contextual dependencies of monitoring parameters and at the same time setting up representative class centers of both normal and abnormal operational states. Consider contrast-enhanced embedding achieved from the previous stage be depicted as $z_i \in \mathbb{R}^p$ where p refers to dimensionality of the learned feature representation corresponding to the i^{th}temporal segment. The objective of the classification step is to assign each embedding $z_i$to a system state label $y_i \in \{0,1\}$, representing normal as well as anomalous conditions, respectively.

First, a prototype-based classification strategy is used to create representative class centers in each of the monitoring conditions. Consider prototype corresponding to class $c$be given in eq. (26), where N_c refers to number of samples belonging to class c.

$$\mu_c = \frac{1}{N_c}\sum_{i=1}^{N_c} z_i \qquad (26)$$

These prototypes are reference anchors depicting the typical structure of any system condition in the embedding space. The similarity between an input embedding and class prototypes is measured by energy-based scoring function as defined in the following eq. (27) with a lower energy value corresponding to a high similarity between input representation and corresponding class prototype.

$$E(z_i, c) = \| z_i - \mu_c \|^2 \qquad (27)$$

The prototype-based energy formulation allows the classifier to set soft decision boundaries between the normal and abnormal monitoring states. To enhance further classification capacity by learning contextual relationships among the temporal embeddings, a transformer-based classifier is incorporated into the framework. Transformer encoder makes global dependencies through a self-attention mechanism that is defined as in eq. (28), where Q, K, and V refers to query, key, as well as value matrices derived from embedding sequence, and d_k refers to dimensionality of the key vectors.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \qquad (28)$$

This attention mechanism allows the classifier to learn relationships between monitoring parameters in various time segments in an efficient manner. Transformer output representation is then added to the prototype-guided energy scores to form the final classification decision in the form of in eq. (29), where $\hat{y}_i$denotes the predicted system condition label.

$$\hat{y}_i = \arg\min_{c} E(z_i, c) \qquad (29)$$

The hybrid classification approach is more robust to noisy monitoring signals and more effective in differentiating between normal and fault-prone operational behaviors. Combining prototype directed energy-based learning and transformer based contextual modeling, the proposed classification phase makes available precise detection of anomalous system states. The estimated system state labels are then sent to the uncertainty-conscience decision-making module with Evidential Deep Learning that enhances the reliability of proactive anomaly notifications in the subsequent phase of the proposed monitoring framework.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

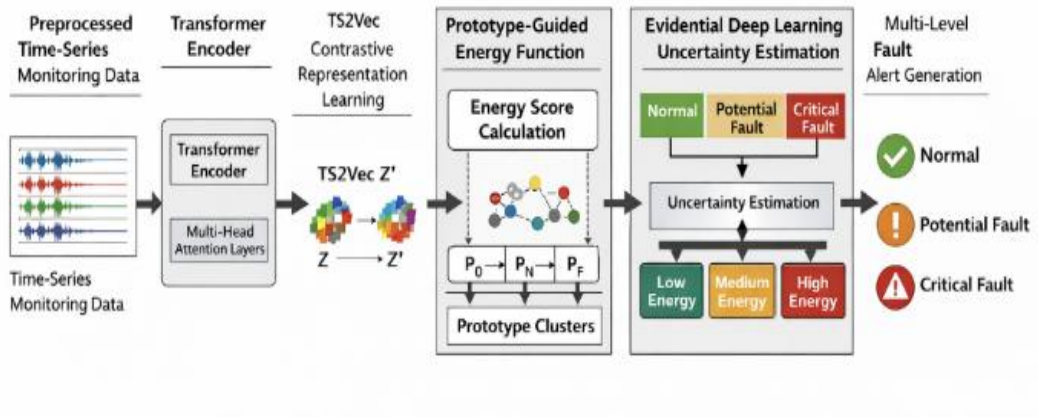


Fig. 2. Architecture of the proposed prototype-guided energy-based transformer classifier for multi-level fault alert generation

The proposed prototype-guided energy-based transformer classifier architecture to generate multi-level fault alerts is presented in Fig. 2.

3.7 Uncertainty-Aware Decision Making Using Evidential Deep Learning

Even though the prototype-guided transformer classifier is an accurate predictor of normal and anomalous system conditions, the real proactive monitoring systems should also measure the confidence of their predictions to minimize false alarms and enhance the reliability of the operation. Thus, the proposed UACP-PM framework includes an uncertainty-conscious decision-making model based on Evidential Deep Learning (EDL), where the model can approximate the confidence of predictions in terms of belief, disbelief, and uncertainty, per each monitoring case. Let the output classifier output of an input embedding be represented as in eq. (30), where $e_c \geq 0$ refers to evidence collected for class c, and C refers to overall system condition classes.

$$f(z_i) = \{e_1, e_2, \ldots, e_C\} \quad (30)$$

The EDL framework has the benefit of representing prediction confidence with a Dirichlet distribution unlike traditional probabilistic classifiers, which simply attempt to directly estimate class probabilities and cannot explicitly represent uncertainty in classification decisions. The Dirichlet distribution parameters are calculated as in eq. (31), where α_c refers to concentration parameter associated with class c. The overall evidence accumulated across all the classes is represented as in eq. (32).

$$\alpha_c = e_c + 1 \quad (31)$$

$$S = \sum_{c=1}^{C} \alpha_c \quad (32)$$

Depending on these parameters, the belief mass assigned to each class is evaluated as in eq. (33), while the total uncertainty mass associated with prediction is evaluated as in eq. (34).

$$b_c = \frac{e_c}{S} \quad (33)$$

$$u = \frac{C}{S} \quad (34)$$

The formulation allows the monitoring framework to differentiate the certainty of anomaly predictions and uncertain classification results due to a lack or inconsistency of evidence in monitoring signals. The Dirichlet probability of each class is calculated as in eq. (35) that gives an uncertainty-conscious estimate of the forecasted condition of the system.

$$P(y = c \mid z_i) = \frac{\alpha_c}{S} \quad (35)$$

Through this evidential representation, the proposed framework enhances reliability of proactive monitoring decision-making and minimizes the chances of false-positive anomaly alerts in the real-life operational conditions.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

3.8 Proactive Alert Generation and Visualization Dashboard

Based on the results of uncertainty-sensitive classification with the help of the Evidential Deep Learning (EDL) module, the proposed UACP-PM framework will produce real-time proactive warnings to help system administrators detect any possible abnormal conditions in time before a critical failure takes place. Assume the uncertainty-aware prediction of classes used in the last step is given as in eq. (36), where $c \in \{0,1\}$ refers to predicted system condition related to normal as well as anomalous operational states, respectively.

$$P(y = c \mid z_i) \tag{36}$$

Depending on probability estimates, proactive alerts get generated by using a predefined decision threshold θ represented in eq. (37), where A_i refers to alert indicator associated with the i^{th}monitoring instance.

$$A_i = \begin{cases} 1, & \text{if } P(y = 1 \mid z_i) \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

On occurrence of the anomaly probability that is above the given threshold value, proactive alert is sent to inform system operators about a possible abnormal behavior in the parameters being monitored. Along with probability-based alert triggering, the uncertainty mass of the EDL module is included into the process of decision-making to enhance reliability of alert generation. Consider uncertainty estimate represented as u_i.

Alerts with high uncertainty scores will be treated as low-confidence predictions and may be re-labeled as requiring further verification, thus improving the false-positive alarm rates in a real-world monitoring setting. The generated alerts are displayed in an interactive visualization dashboard.

4. RESULT AND DISCUSSION

In this section, the experimental findings are provided based on the application of the proposed UACP-PM framework to detect anomalies proactively in the data of IT systems monitoring. The standard preprocessing validation, classification metrics, convergence analysis, confusion matrix evaluation, ROC curve analysis, alert distribution study, feature contribution ranking and comparative performance analysis with the existing methods are used to determine the performance of the framework. The findings indicate that the proposed solution is effective in detecting the normal and abnormal states of the systems with a high level of accuracy, with minimal false alarms.

4.1 Dataset Label Distribution Before and After Class Balancing

Fig. 3 shows the class distribution of the network anomaly detection dataset prior to balancing process and after balancing. First, the dataset had a large class imbalance with the normal class (label 0) having 4398 samples and the anomaly class (label 1) having 600 samples. This imbalance may have a harmful impact on model training with the tendency to predict the majority class and decrease the sensitivity to anomalies. To deal with this problem, there was a class balancing approach to balance the number of samples in each category.

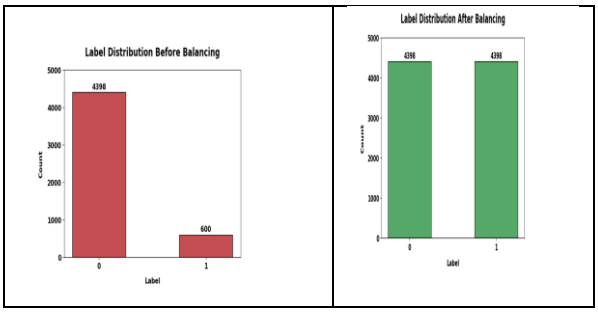


Fig. 3. Label distribution of the dataset before and after class balancing showing correction of class imbalance between normal and anomalous samples

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

4.2 Impact of Outlier Removal on Feature Distribution

Fig. 4 gives the distribution of the ifInOctets11 feature before and after the outlier removal process used during the data preprocessing. Before preprocessing, the feature had uneven extreme changes, which may adversely affect the convergence of the model and lower the quality of the performance of the anomaly detector. Measurement noise, transmission bursts, or abnormal system behavior not related to persistent faults are common causes of an outlier in network traffic metrics. In order to enhance the quality of the dataset, an outlier removal method was used to remove the gross deviations whilst maintaining the necessary statistical properties of the feature distribution.

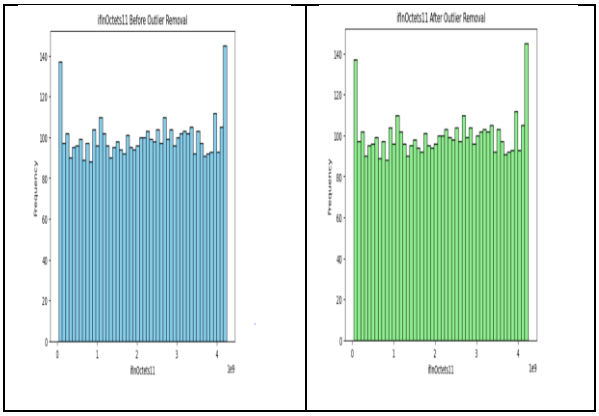


Fig.4. Distribution of the *ifInOctets11* feature before and after outlier removal illustrating improved data consistency for model training

4.3 Layer-Wise Configuration of the Proposed Model Architecture

Fig. 5 shows the layered architecture of the proposed proactive monitoring system, which shows how input time-series measures are transformed with deep learning layers. The architecture normalizes sequential system parameters using temporal feature extraction modules to learn short term fluctuations and long-term dependencies in the behavior of an IT system. The refined embeddings are then sent to classification layers that identify normal and anomaly system states.

Layer (type)	Output Shape	Param #	Connected to
input_layer_3 (InputLayer)	(None, 30, 320)	0	-
multi_head_attention_1 (MultiHeadAttention)	(None, 30, 320)	1,642,560	input_layer_3[0][0], input_layer_3[0][0]
dropout_5 (Dropout)	(None, 30, 320)	0	multi_head_attention_1[0]...
add_2 (Add)	(None, 30, 320)	0	input_layer_3[0][0], dropout_5[0][0]
layer_normalization_2 (LayerNormalization)	(None, 30, 320)	640	add_2[0][0]
sequential_1 (Sequential)	(None, 30, 320)	41,344	layer_normalization_2[0][...
dropout_6 (Dropout)	(None, 30, 320)	0	sequential_1[0][0]
add_3 (Add)	(None, 30, 320)	0	layer_normalization_2[0][... dropout_6[0][0]
layer_normalization_3 (LayerNormalization)	(None, 30, 320)	640	add_3[0][0]
global_average_pooling1d_1 (GlobalAveragePooling1D)	(None, 320)	0	layer_normalization_3[0][...
dense_6 (Dense)	(None, 128)	41,088	global_average_pooling1d_...
dropout_7 (Dropout)	(None, 128)	0	dense_6[0][0]
dense_7 (Dense)	(None, 2)	258	dropout_7[0][0]

Fig.5. Layer-wise summary of the proposed deep learning architecture showing output dimensions and sequential feature transformation across temporal representation and classification stages

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

4.4 Classification Performance Evaluation of the Proposed Model

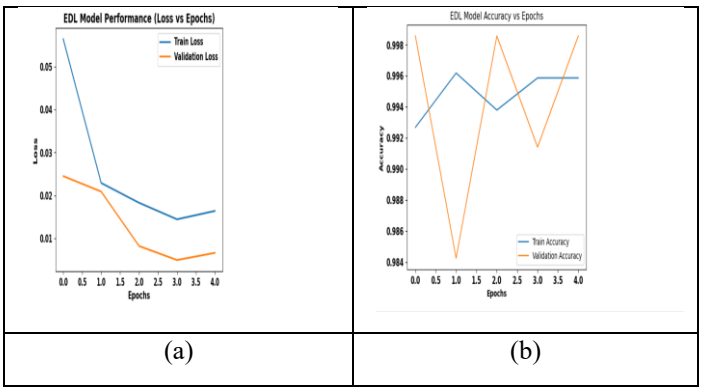
Table 2 shows the performance of the proposed proactive monitoring framework in terms of classification with the standard performance measures such as precision, recall, F1-score, and overall accuracy. The model attained both a precision of 0.98 and recall of 0.98 with both normal and anomalous classes, which implies that the model is highly effective in identifying system states, with a very low false positive and false negative. The value of F1-score, 0.98, also indicates the balanced accuracy between detection sensitivity and prediction reliability between the two classes. The fact that the proposed deep learning-based framework classifies the overall accuracy of 98% proves that the framework is effective in capturing temporal patterns and distinguishing between abnormal and normal operational conditions of the system.

Table 2: Classification performance metrics of the proposed proactive monitoring model

Class	Precision	Recall	F1-score	Support
0	0.98	0.98	0.98	868
1	0.98	0.98	0.98	880
Accuracy	-	-	0.98	1748
Macro avg	0.98	0.98	0.98	1748
Weighted avg	0.98	0.98	0.98	1748

4.5 Training Convergence Analysis of the Proposed Framework

As shown in Fig. 6(a) the curves of training and validation loss indicate a steady decrease throughout the epochs, which indicates a successful optimization of the proposed uncertainty-aware proactive monitoring framework.. Fig. 6(b) shows that the training accuracy stays high and constant across the epochs, whereas the validation accuracy does change significantly.



(a) (b)

Fig. 6. Training and validation performance of the EDL model: (a) Loss vs Epochs, (b) Accuracy vs Epochs

4.6 Confusion Matrix-Based Performance Evaluation

The confusion matrix in Fig. 7 shows the classification results of the proposed model. All the 854 normal cases and all the 862 abnormal cases are properly classified and only 14 normal samples are wrongly classified as abnormal and 18 abnormal samples as normal.. Such a trade-off in reducing false positives and false negatives, suggests that the model is not only highly accurate but also reliable in identifying anomalies.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

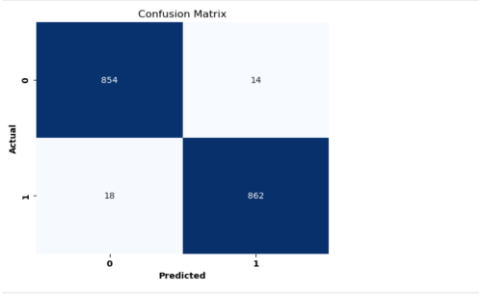


Fig. 7. Confusion matrix of the EDL + Energy-based classifier for normal (class 0) and abnormal (class 1) states

4.7 ROC Curve and AUC Performance Evaluation

Fig. 8 shows the ROC curve of the EDL model, demonstrating the discriminative capabilities of the model to differentiate normal and abnormal states of the system.. The Area Under the Curve (AUC) of 0.999 supports the fact that the two classes are almost perfectly separated. The outcome indicates that the model has a very high true positive rate and a low false positive rate.

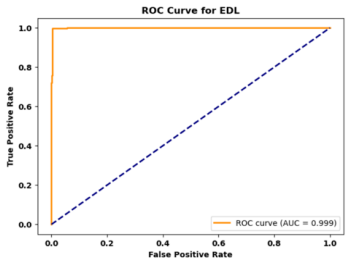


Fig. 8. Receiver Operating Characteristic (ROC) curve of the EDL model with AUC = 0.999

4.8 Class-wise Performance Comparison

Fig. 9 shows the performance of the proposed model by class using the metrics of precision, recall, and F1-score. The normal and fault classes have very high values of 0.98 in all the three measures. The balanced scores also demonstrate the strength of the framework, and there is no preference to a particular class at the cost of the other. A high level of precision lowers the rate of false alarms and a good recall ensures that the true anomalies are well represented.

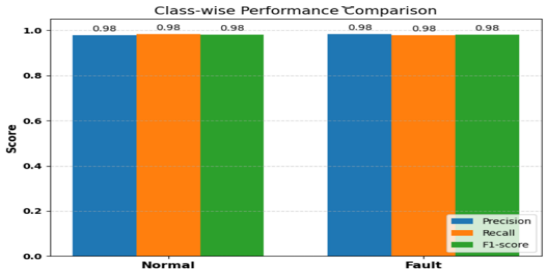


Fig. 9. Class-wise comparison of precision, recall, and F1-score for normal and fault classes

4.9 Proactive Alert Distribution Analysis

Fig. 10 shows how proactive alerts are distributed in three categories, namely, normal, potential fault and critical fault. There are 837 normal system behavior alerts, 736 alerts that are potential faults, and 175 alerts that are critical faults. This distribution illustrates the fact that the framework can distinguish between different degrees of system risk.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

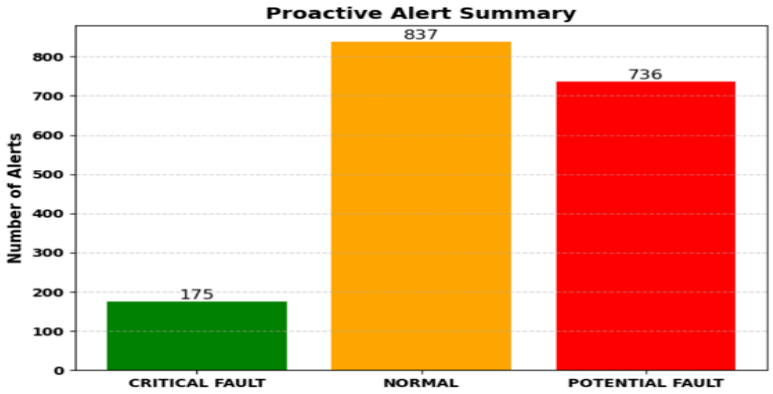


Fig. 10. Summary of proactive alerts generated by the proposed framework, categorized into normal, potential fault, and critical fault conditions

4.10 Key Parameters Driving Fault Alerts

Fig. 11 lists the ten most contributing parameters in terms of their contribution on the number of fault alerts. Feature_15 emerges as the most influential parameter (0.0200), followed by Feature_10 (0.0151) and Feature_11 (0.0142). Other notable ones are Feature_9, Feature_29 and Feature 27 with values of more than 0.0130 and Feature 16 and Feature 21 with comparatively low yet important values. This ranking brings into focus the parameters that dominate in raising fault alerts.

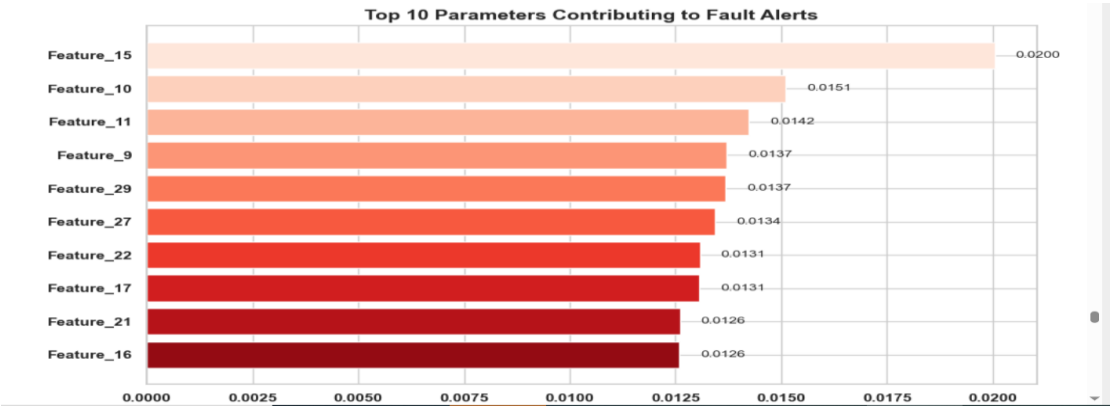


Fig. 11. Top 10 parameters contributing to fault alerts ranked by their relative importance

4.11 Comparative Model Performance Evaluation

Fig. 12(a) shows that Prototype Energy classifier has a moderate performance with an accuracy of 0.918 and F1-score of 0.922. Although the precision (0.925) is a bit higher, the recall (0.915) shows that there are cases where faults are not detected adequately, which is a weakness in detecting anomalies. In Fig. 12(b), the Transformer-based classifier demonstrates better performance with a 0.946 accuracy and a balance between the precision and the recall of 0.945. This advancement underscores the fact that transformer architectures are able to better represent contextual dependencies between monitoring parameters. The Transformer + Energy model, as in Fig. 12 (c), has a further improvement in its performance with an accuracy of 0.972 and F1-score of 0.973.. Lastly, the EDL + Energy classifier also achieves the best overall performance with accuracy (0.982), F1-score (0.982), precision (0.981), and recall (0.980) in Fig. 12(d). Evidential Deep Learning also allows making uncertainty-aware decisions, which guarantee effective predictions with minimal false alarms.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

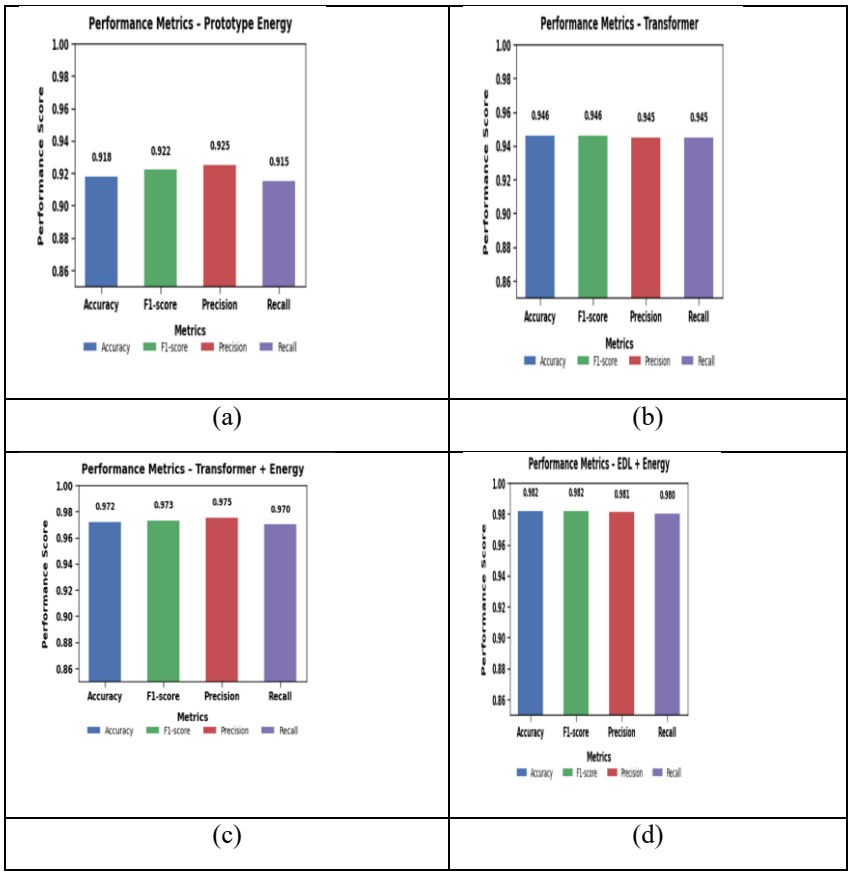


Fig. 12. Performance comparison across different classifiers: (a) Prototype Energy, (b) Transformer, (c) Transformer + Energy, and (d) EDL + Energy

According to this comparative analysis, the EDL + Energy classifier is chosen as a final decision module of the suggested proactive monitoring framework as it is superior in classification accuracy, balanced precision-recall operation, and uncertainty-aware prediction ability.

4.12 Comparative Performance Analysis with Existing Methods

Table 3 provides a comparative analysis of the performance of the proposed uncertainty-aware proactive monitoring framework and some of the traditional machine learning and deep learning models, such as SVM, Random Forest, CNN, LSTM, Transformer, and energy-based classifiers. Conventional machine learning approaches like SVM and Random Forest exhibit moderate performance in classification but exhibit weaknesses in the ability to capture temporal relationships that are manifested in system monitoring data. Deep learning models like CNN and LSTM can enhance their performance by learning hierarchical and sequential representations, but their performance in presenting long-range contextual dependencies is not as effective. The Transformer-based classifier also works to increase the performance of the classifier, by effectively capturing global temporal relationships among monitoring parameters. The energy-based classification also helps to strengthen decision boundaries. Lastly, the suggested Evidential Deep Learning (EDL) with the Energy-based classifier exhibits the best performance on all evaluation metrics and has an accuracy of 98.2% and balanced values in terms of precision and recall as well as F1-score.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 21 January 2026 **Revised**: 12 March 2026 **Accepted**: 14 April 2026

Table 3: Performance comparison of the proposed framework with existing machine learning and deep learning methods

Method	Accuracy	Precision	Recall	F1-score
SVM	0.902	0.895	0.887	0.891
Random Forest	0.931	0.928	0.921	0.924
CNN	0.944	0.941	0.938	0.939
LSTM	0.956	0.952	0.948	0.950
Transformer	0.946	0.945	0.945	0.945
Prototype Energy Classifier	0.918	0.925	0.915	0.922
Transformer + Energy	0.972	0.971	0.970	0.973
Proposed DL+ Energy Model	0.982	0.981	0.980	0.982

This enhancement validates that uncertainty-aware learning with hybrid representation modeling is instrumental in boosting the dependability of anomaly identification and can be used to help maintain robust proactive observation of key IT system parameters.

4.13 Discussion

The experimental findings reveal that the proposed proactive monitoring framework is effective in identifying anomalies in the parameters of IT systems at high accuracy and reliability. Both the training and validation loss curves depict that the model converges stably and the model exhibits desirable learning behavior with no major overfitting.

The effective ability of the proposed model in discriminating between normal and fault conditions is further confirmed by the ROC curve that attains an almost perfect separation between the normal and fault conditions. The obtained high precision, recall, and F1-score values on the two classes demonstrate balanced classification performance, which is crucial in reliable anomaly detection in proactive monitoring systems.

Moreover, the analysis of the alert distribution reveals that the framework is effective in classifying system conditions into normal, potentially fault, and critical fault level. The comparative analysis of the proposed framework against the models of machine learning and deep learning reveals that the proposed model is better performing. Transformer-based learning, energy-based classification, and uncertainty-aware decision-making contribute to the overall performance of detecting anomalies and minimizing the rates of false alarms.

4.14 Conclusion

In this paper, the UACP-PM was introduced to identify anomalies in critical parameters of the IT system. The presented framework combines the use of temporal feature extraction, contrastive representation learning, prototype-guided energy-based classification, and evidential deep learning to enhance the performance of anomaly detection. The experiments indicate that with the proposed model, the classification accuracy was 98.2%, with 98.1% precision and 98.0% recall, and 98.2% F1-score. The AUC of ROC analysis was also obtained as 0.999, which revealed the high classification ability. Besides that, the framework was able to come up with graded alert levels of normal, potential fault and critical fault conditions, which facilitated consistent proactive monitoring. The proposed UACP-PM framework can be scaled in future work by integrating real-time streaming information and dynamic online learning processes to enhance further the scaling and performance to dynamic IT monitoring conditions.

REFERENCES

[1] Somma, M., Gallien, T., & Stojanovic, B. (2025). Anomaly detection in complex dynamical systems: A systematic framework using embedding theory and physics-inspired consistency. *arXiv preprint arXiv:2502.19307*.

[2] Ugolini, A. R., Leoni, J., Breschi, V., Paniccia, D., Tucci, F. A., Capone, L., & Tanelli, M. (2026). Explainable condition monitoring via probabilistic anomaly detection applied to helicopter transmissions. *Engineering Applications of Artificial Intelligence*, *173*, 114420.

[3] Chourasiya, L., Khatri, S., Lilhore, U. K., Simaiya, S., Alroobaea, R., Baqasah, A. M., ... & Khan, M. (2025). Advanced system log analyzer for anomaly detection and cyber forensic investigations using LSTM and transformer networks. *Journal of Cloud Computing*, *14*(1), 60.

[4] Ortiz-Garcés, I., Villegas-Ch, W., & Luján-Mora, S. (2025). Autonomous cyber-physical security middleware for IoT: anomaly detection and adaptive response in hybrid environments. *Frontiers in Artificial Intelligence*, *8*, 1675132.

[5] Xin, R., Liu, H., Chen, P., & Zhao, Z. (2023). Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework. *Journal of Cloud Computing*, *12*(1), 7.

[6] He, H., Li, X., Chen, P., Chen, J., Liu, M., & Wu, L. (2024). Efficiently localizing system anomalies for cloud infrastructures: a novel dynamic graph transformer based parallel framework. *Journal of Cloud Computing*, *13*(1), 115.

[7] Jiang, G. (2024). Artificial intelligence-based adaptive anomaly detection technology for IaaS cloud virtual machines. *Journal of Engineering and Applied Science*, *71*(1), 102.

[8] Gu, W., Zhong, R., Yu, G., Sun, X., Liu, J., Huo, Y., ... & Lyu, M. R. (2026). KPIRoot+: An efficient integrated framework for anomaly detection and root cause analysis in large-scale cloud systems. *Empirical Software Engineering*, *31*(2), 28.