Adaptive Deep Learning Framework with Unicintus Optimization for Anomaly Detection in Streaming Data

Srividhya V R[1], Kayarvizhy N[2]
Computer Science and Engineering, B.M.S. College of Engineering,
Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India-590018[1, 2]
Computer Science and Engineering, RV Institute of Technology and Management, Bangalore, Karnataka, India[1]

***Abstract*—Anomaly detection in streaming data is crucial for identifying unusual patterns or outliers that may indicate significant issues. Traditional methods struggle with the inability in efficiently handling high-velocity data, adapting to changing data distributions, and maintain performance over time. Further, the conventional methods struggled with scalability, adaptability, and computational efficiency, leading to delays in detection or an increased rate of false positives. To address these limitations, Unicintus Escape Energy enabled Sampling based Drift Deep Belief Network-Bidirectional Long Short Term Memory (UES2-DTM) is proposed in the research. The research model incorporates the combination of adaptive reservoir sampling as well as the adaptive sliding window mechanisms into the base model, which elevates the efficiency of the model to work with the streaming data. Moreover, the adaptive sliding window mechanisms for drift detection integrates the Unicintus Escape Energy Optimization (UE2O) Algorithm to boost efficiency by dynamically adjusting the sliding window size and parameters, based on real-time streaming data characteristics. Further, Adaptive reservoir sampling helps in maintaining a representative sample of the data stream, for effective detection. Overall, the UES2-DTM model demonstrates superior adaptability and accuracy, which is evaluated with the metrics such as precision, recall, F1-score, and Mean Square Error (MSE) attained 97.199%, 94.827%, 95.998%, and 3.461 respectively.**

Keywords*—*Streaming data; sliding window; anomaly detection; reservoir sampling; Unicintus escape energy optimization

I. Introduction

Due to its ongoing use in real-world issues, the Internet of Things (IoT) has gained increasing relevance in recent years [1-4]. As a result of this advancement, the internet expanded and was rolled out across several devices, resulting in rapid global growth. IoT is a key component in computing systems that enable intelligent data collection and analysis even in the absence of known entities [5-6]. Road traffic, supply chain management [7], healthcare, smart cities [8], transit, and more were all made possible by the Internet [9-10]. However, there are several restricted qualities of this independent object, such as minimal memory, a small CPU, low bandwidth channels for communication, and so forth [9]. Large datasets were also needed for analysis and decision-making in the logistics operation [10]. When data from several IoT sensors combine to create complex patterns, sometimes known as unique events [10], the result is an anomaly [4]. The scenario is hazardous since these rare and complex events have undesired data and barely happened. The complex event processing (CEP) approach was created to process, evaluate, and summarize complicated events [11] [10]. One effective component of web-based apps that increases the ability to predict complex events and anomalous activity in real-time data streams is the CEP.

Additionally, in a gamut of applications such as monitoring of traffic congestion, live mapping, smart street lamps, and so forth [12-15], the CEP method can also effectively predict the real-time streaming data with the sequence of events and suspicious actions [16-18]. This aids in the development of automated systems for smart cities [19] [10]. The CEP was more adaptable and could make excellent use of a significant amount of continuously streamed data to facilitate decision-making. The traffic congestion management system's forecast resulted in significant changes to traffic patterns, including shorter travel times, more road capacity, and the elimination of fuel usage and air pollution [20]. By linking all parts to the central server, the message unit accurately predicts data via wireless networks using Internet of Things sensors, enabling the traffic congestion prediction to function as intended [21]. Utilizing efficient deep learning (DL) and machine learning (ML) approaches, which offer greater benefits for efficient event prediction, improved IoT-based congestion prediction in various industries. To produce predictions, both structured and unstructured data may be used with the very effective ML and DL algorithms [22-23].

By removing the useful streaming data from the IoT sensors, several ML and DL techniques improved the prediction process and produced very accurate predictions. With the aid of the Markov decision process model, the Bayesian network was a q-learning technique that was used to anticipate future occurrences in advance [6]. In addition to having the capacity to produce dynamic and scalable anticipated results, ML-based approaches such as SVR [24], DT [25] are more dependable than other approaches that train the models using historical data [10]. DL approaches effectively overcome the constraints of ML, even though the prediction requires very complicated and high-quality data. Anomaly detection in networks involves identifying unusual patterns in network traffic, often referred to as anomalies or outliers. These nonconforming patterns have applications in fraud detection, cyber security, and military surveillance. For instance, anomalous traffic patterns may indicate sensitive data being sent to unauthorized hosts [26].

Thus, to address and tackle the described challenges, the UES2-DTM model is proposed in the research.

The research model UES2-DTM aims to work with anomaly detection from the streaming data. The research model obtains efficient outcomes with the combination of contributed mechanisms that enhanced the reliability and scalability of UES2-DTM. In addition, the preprocessing and feature extraction mechanisms aid in obtaining significant outcomes specifically when working with the streaming data. A novel approach, encompassing the following is introduced in this work:

- Develop an Adaptive Reservoir Sampling technique to effectively handle large-scale, high-velocity data streams with unknown total sizes.
- Design an Adaptive Sliding Window-Based Drift Detection mechanism enhanced by the Unicintus Escape Energy Optimization (UE2O) algorithm. This approach aims to dynamically adjust to data distribution changes, improving the precision of anomaly detection in streaming data.
- Construct a hybrid deep learning framework combining Deep Belief Networks (DBN) and Bidirectional Long Short-Term Memory (BiLSTM) networks. This model will incorporate the proposed sampling and drift detection techniques to strengthen the IoT data streams anomalies.
- Assess the proposed UES2-DTM model using relevant metrics. This evaluation will benchmark the model's effectiveness against existing methods in detecting anomalies within streaming data environments.

The research article is organized as, Section II describes the Related Work and Section III elaborates on the system modeling of the UES2-DTM. Section IV analyzes the research outcomes, and section V ends the research with suggestion for future work.

II. Related Work

The existing research on the anomaly detection with the live streaming data is elaborated in this section. The Seasonal Auto-Regressive Integrated Moving Average (SARIMA) [23] and Bidirectional Long Short-Term Memory (Bi-LSTM) [23] were first presented by Ayushi Chahal et al.. The intent of this research model was to improve inhabitants' quality of life. Any type of time-series dataset could be employed with the suggested approach, including forecasting stock trends, diseases, and weather patterns. The research might also focus on improving the suggested model prediction performance through the use of various interpretation analysis techniques.

The online event anomaly detection with XGBoost, LSTM, and RF was first presented by Suhwan Lee et al. [27]. The suggested method retrained the model using the most current cases that were recently recorded on the event stream via a sliding window. There exist several issues with research that still require attention. An occurrence that was deemed abnormal could be reclassified as it failed to update the forecast. Nevertheless, it could be highly instructive to consider such modifications that justified the anticipated anomalies and could enhance model performance.

The Preprocessed Isolation Forest (PiForest) technique for anomaly identification was initially described by Prarthi Jain et al. [10]. The method referred to as the PiForest was applying the iForest algorithm to datasets that were drastically decreased in dimensionality. To handle such data and efficiently detect anomalies, a sliding window was employed in the research. The method's effectiveness in identifying anomalies could be confirmed by contrasting its results with the output of many established anomaly detection algorithms.

A deep neural network (DNN) was presented by Asmaa F. Hassan et al. [28] to address the outlier detection issue with the streaming data input. The experiment's findings showed that it achieved superior detection accuracy with a minimal false alarm rate than two cutting-edge DL techniques. However, the present stage of the suggested approach was not entirely inadequate due to the length of time needed to train the system and its exclusive focus on finding global outliers. The multiclass classification scenarios could be included to tackle the outlier identification problem more successfully. The issue of contextual outliers could be discussed to improve the research efficacy.

The recurrent neural network (RNN) model, which was presented by Jun Liu et al. [21], not only lowers regression error but also has the ability to identify anomalous data that was acquired by IoT terminal nodes ensuring that network predictions were robust and stable. To enable prompt repair and management of sensor nodes, the system would get feedback when there are medium- and long-term irregularities.

A framework based on isolation forests with dynamic Insertion and Deletion methods (IDForest) was presented by Haolong Xiang and Xuyun Zhang [29]. By gradually learning the tree structure, IDForest quickly and accurately identified abnormalities in the data stream containing large amounts of data. Additionally, edge computing investigations confirmed that deploying in parallel, increased detection speed by hundreds of times. The research model could implement edge computing settings. Further, the research model did not work with noise reduction that could be implemented to improve efficacy.

Cube sampling and the iForest algorithm were first presented by Seemandhar Jain et al. [30] as methods for identifying anomalies. The use of sliding windows to handle such data remained efficient. The effectiveness of the approach in identifying anomalies was exhibited by a comparative analysis with several widely recognized anomaly detection algorithms. Still, the handling of streaming data was not performed well in the research. The Online evolving Spiking Neural Network (OeSNN) classifier [31] was presented for anomaly detection by Piotr S. Maciąg et al.. OeSNN-UAD did not divide output neurons into decision classes that are predetermined. Rather, every newly formed output neuron on OeSNN-UAD was given an output value, which was determined at random using the most recent input values. Nevertheless, the model was unsuitable for settings with stringent memory constraints. A comparative analysis of other existing approaches is depicted in Table I.

TABLE I. LIMITATIONS OF EXISTING APPROACHES AND OPTIMIZATIONS

Existing Approaches	Limitations
ARIMA	Limited Drift Detection capability, Struggles with non-stationary data
Rule-Based Approaches	Limited Adaptability, High false-positive rate
Clustering	Can struggle with high-dimensional data, Requires prior knowledge of clusters
Supervised Machine Learning	Needs extensive labelled data, struggles with drift, slow inferences in real time prediction
Convolutional Neural Networks	Not well-suited for sequential/temporal anomaly detection
Long Short Term Memory	Struggle to quickly adapt to shifts in data distribution without retraining or fine-tuning, more like a back box so interpretability is difficult, not immune to vanishing gradient problem
Genetic Algorithm	Slow convergence, relies on fixed evolutionary operations like mutation and crossover.
Particle Swarm Optimization	Can get stuck in local optima, Prone to premature convergence in complex problems
Bayesian Optimization	Assumes a stationary function landscape, limiting adaptability to non-stationary data

III. PROPOSED SYSTEM MODEL

A. Anomaly Detection with Unicintus Escape Energy Enabled Sampling Based Drift Deep Belief Network-Bidirectional Long Short Term Memory Model

The research to detect anomalies in streaming data is performed with the UES2-DTM model. The research is initiated with the streaming data that acts as the input. The streaming Apache Kafka system is used to obtain streaming data that involves information from various sectors. The obtained inputs from different sectors are aggregated with the data aggregator, which is available in the data aggregation block. Once data are aggregated, the input is fed into the preprocessing block, where the missing data imputation and the logarithmic data sampling take place with K-nearest neighbor (KNN) and logarithmic normalization respectively.

The preprocessed data serves as the input for the Feature Extraction block, aiming to identify and extract the most relevant features. To obtain efficient outcomes of feature extraction, statistical features, time-series informative features, and mixed information metric features are extracted in the research. The combined outcome represented as the feature vector is fed into the model UES2-DTM, which is hybridized with the adaptive reservoir sampling as well as the adaptive sliding window-based drift detection mechanism. The mechanisms involved are optimized with the hybrid optimization that integrates the characteristics of Rabbit and Harris Hawk. The adaptive reservoir. sampling splits the data into sub-sets that enhance the processing time by neglecting the entire data processing. Further, the adaptive sliding window-based drift detection mechanism obtains information on the drift in the limited frames achieved at the typical time frames obtained through the sliding window process. The UE2O algorithm in the research aids in tuning the described process to obtain the optimal outcomes. The entire workflow of the research is shown in Fig. 1.

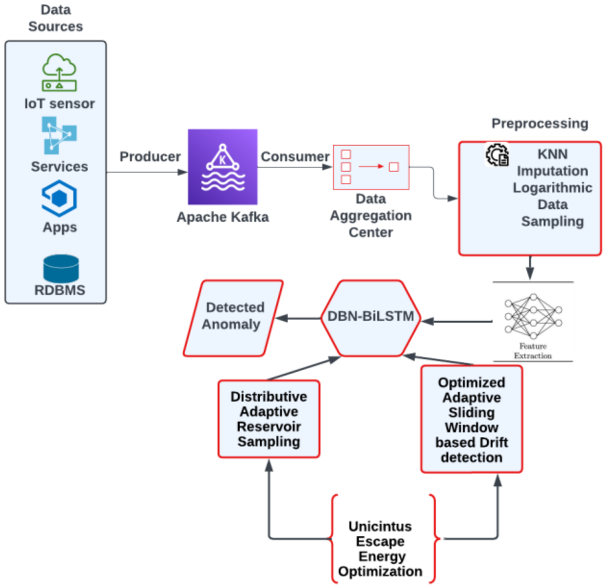


Fig. 1. Block diagram of the proposed methodology.

B. Input Streaming Data

The input streaming data is obtained from the Kafka streaming software that collects the data from different data sources at different time intervals. The data from data sources are forwarded from the producers to the Apache Kafka system, which is sent to the consumers based on their requirements. The input streaming data from the Kafka streaming is represented as,

$$S_{data} = \{\ldots.,S^{v-1},S^{v},S^{v+1},\ldots..\} \quad (1)$$

where, $S^{v}$is the data instance at time v, $S^{v-1}$and S^{v+1} are the previous and the next data instances.

C. Preprocessing with Imputer and Logarithmic Data Sampling

Preprocessing is performed in the research to achieve the most promising data that aids in obtaining accurate outcomes in further process. The preprocessing is performed with the missing data imputation and logarithmic data sampling. The missing data imputation is performed with the KNN imputer. Due to the efficacy in solving the issues with the data imputation, KNN is chosen as the imputer, which further works without the intervention of detection models. The popular Euclidean distance equation is used for the above [32]. With the outcome of missing value imputed, logarithmic data sampling [33] is performed in the research.

D. Feature Extraction with Time-series Statistical Mixed Information Features

The feature extraction is performed to retrieve the features of the preprocessed data, for which the time-series informative features, statistical features, and mixed information metric features are utilized in the research.

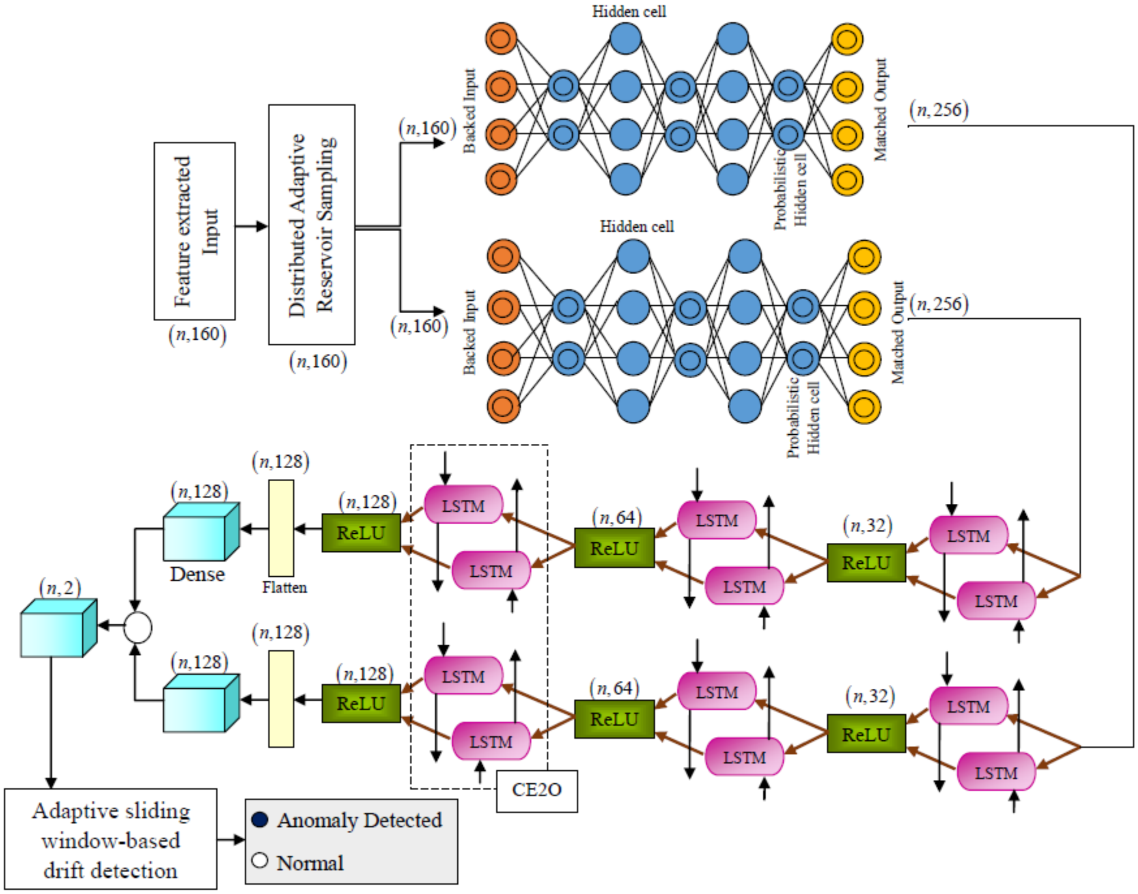


Fig. 2. Architecture of UES2-DTM model in anomaly detection.

1) Time-Series informative features: The time series informative features are derived with the help of the TSFEL library. The TSFEL library works with over 65 different features based on the temporal, spectral, statistical, and fractal domains. The statistical domain includes the time-series information concerning mean, variance and so on. Further for a temporal domain, features such as autocorrelation, and centroid are considered in the research.

2) Statistical features: The evaluated statistical features in the research of anomaly detection with the streaming data are mean, median, standard deviation, variance, skewness, kurtosis, range, and interquartile range (IQR). The obtained statistical features are concatenated to form the feature vector.

3) Mixed information metric features: The mixed information metric feature is estimated with mutual information that extracts features with Shannon entropy, and the new feature is evaluated based on the probability density function (PDF) that involves the Parzen Window method.

E. Adaptive Reservoir Sampling

Adaptive Reservoir Sampling is an advanced technique designed to manage large or streaming data sets by maintaining a representative sample of the data. This method is particularly useful when dealing with data streams, where the total size is unknown or too large to store in memory.

At the start, a fixed size reservoir is initialized to hold a sample of data points [35]. This reservoir is typically filled with the first I elements from the stream in which I is the size of the reservoir. As new data points arrive in the stream, they need to be considered for inclusion in the reservoir. For each new data point, a random decision is made to either include the new point in the reservoir or replace an existing point. The probability of replacing an existing point is proportional to its index in the stream, ensuring that each data point has an equal chance of being included in the final sample. Specifically, for a stream index O and reservoir size I, the probability of a new element new replacing an existing element in the reservoir is I/O. This ensures that each element in the stream has an equal likelihood of being in the reservoir by the end of the process. The explained execution takes place when the reservoir size remains unchanged. The major advantage exhibited in the research model is the adaptive mechanism that adjusts the size of the reservoir dynamically based on data distribution. This adaptive behavior ensures that the reservoir reflects the most relevant data characteristics. Thus, the reservoir size is strengthened and impaired accordingly, to evaluate the efficiency. If the size of the reservoir is decreased by ρ, then the number of elements are neglected from the reservoir and continue to work as the traditional reservoir sampling. If the reservoir size is increased by ρ, then the minimum value of incoming elements ele_{min} is evaluated that possess the uniformity coincidence to exceed the threshold value Th. Further, the algorithm is flipped to attain the number of elements kr to retain among the total elements I. The probability of kr is estimated as,

$$p(kr) = \frac{\binom{o}{kr}\binom{ele_{min}}{I+\rho-kr}}{\binom{o+ele_{min}}{I+\rho}} \quad (2)$$

F. Unicintus Escape Energy Enabled Model

The research on anomaly detection from the streaming data is performed with the UES2-DTM, which has the baseline model DTM that combines the DBN as well as the BiLSTM. Though the provided baseline models are advanced neural networks, they exhibited certain drawbacks in individual working areas. Hence the combination of them along with UE2O algorithm is proposed in the research of anomaly detection specifically with the streaming data that overcomes the drawbacks of both simultaneously and provides highly efficient outcomes.

DBNs are probabilistic graphical models consisting of multiple layers of stochastic hidden variables., which are built from Restricted Boltzmann Machines (RBMs) stacked on top of each other, followed by a fine-tuning step with a supervised classifier [34]. Hence, In DBN there exist several hidden layers to process the outcome. DBNs expose the advantages of learning the hierarchical representations of the input data that in addition aids in capturing the complex patterns and structures. Further, DBNs exhibit the behavior of dimensionality reduction while preserving the significant features of the input. The BiLSTM network acts as the extension of the Long Short-Term Memory (LSTM) that processes each data in both forward and backward directions intending to capture the long-term dependencies as well as the temporal patterns. Thus, the integration of both emerges the highly efficient research model, where the DBN extracts the most promising features followed by the BiLSTM that analyzes them over time to understand the past and future behaviors, which detects the deviations or anomalies accurately. The advantages of DTM are highly efficient in terms of anomaly detection, even though the detection in the streaming data remains crucial. Thus, the adaptive reservoir sampling mechanism as well as the UE2O-optimized adaptive sliding window-based drift detection mechanism is integrated with the DTM. The working model of the UES2-DTM is depicted in Fig. 2.

The integration of escape characteristics of rabbits [37], and hunting energy characteristics of Hawks [38] forms the UE2O algorithm. Rapid, unpredictable movements made by the rabbit to avoid predators serve as a metaphor for the necessity of flexible, responsive anomaly detection methods. However, the characteristics of hawks, who are renowned for their well-thought-out, highly effective hunting tactics. The input streaming data for anomaly prediction is represented in (1). The data is partitioned into equal-length subsets using a sliding window of size m:

$$E_v = \{S_v, S_{v-1}, \ldots\ldots S_{v-m}\} \quad (3)$$

where, $E_v$is the new data instance with m dimensions that represent the original state at the time v on the data stream which is fed into UES2-DTM for anomaly detection. To address concept drift, an adaptive sliding window updates the threshold dynamically:

$$E_v(new) = E_{vlow} + \left|E_{vlow} - E_{vup}\right|\gamma \quad (4)$$

where, $E_v(new)$is the new data generated from each data stream, γ is the adaptive factor, E_{vlow} is the lower bound, and E_{vup} is the upper bound of the data level. The solution positions are randomly initialized within search bounds. The objective function is:

$$M\left(E_v(new)\right) = max\left(accuracy\left(E_v(new)\right)\right) \quad (5)$$

where, M indicates the objective function .Unicintus Optimization aims to maximize this objective function using a hybrid approach.

The core idea behind the hybrid optimization technique is to prove the performance of it in non-stationery environments like IoT anomaly detection. Artificial Rabbit Optimization (ARO) is used for global exploration whereas Harris Hawk Optimization (HHO) is used for local exploitation to merge the advantages of both. ARO is used to find an initial good weight candidate (that helps us explore broadly) whereas HHO is used to refine that candidate weight for better accuracy (helps to fine tune for the optimal performance). Eventually the final optimized weight are arrived at. Accurate drift detection in an adaptive sliding window framework is made possible by the research's efficiency. Furthermore, the framework preserves the overall efficiency of the anomaly detection model.The working model of the UES2-DTM initiates with the feature extracted output. The input is fed into the adaptive reservoir sampling mechanism, where the sample data are held accurately and proceeded to avoid data collision as well as the overfitting issue. The outcome fetched from the reservoir sampling process is further fed into the DBN network, where the features are extracted accurately. With the mechanism applied at the adaptive reservoir sampling the further process works in two consecutive sections. The outcome of both DBNs is provided as the input to the BiLSTM, where three simultaneous BiLSTMs are connected together in each parallel row. This outcome is fed into the flattened layer followed by the dense one. Moreover, the outcome of dense layers at each parallel row is concatenated and presented into the dense layer, where the outcome as normal and anomaly detected is achieved in the work.

G. Adaptive Sliding Window-Based Drift Detection

Sliding Window Drift Detection is a method used to identify changes or drifts in data distributions over time, particularly in streaming data environments, which helps in monitoring and adapting to shifts in data patterns that may affect model performance. It helps to detect and identify a time instant (or interval) when a change arises in the new data. The drift detection helps to evaluate the model’s reliability. A fixed-size window is a critical parameter that affects sensitivity and detection performance. Thus, the window slides over the data stream, continually updating its position as new data points arrive [36]. As new data points arrive, they are incorporated into the current window. Each window data point is evaluated against the error metrics, out of all the maximum error that occurred is considered as the threshold. The maximum error of each window is declared through the fitness of the UE2O algorithm. With the obtained threshold, the next iteration is evaluated and updated, hence called as adaptive in the proposed mechanism. The current window's statistical measures are compared with those from previous windows to detect the drift. The statistical measures represent the Threshold value of the drift. Thus, the maximum error is declared drift and on the next iteration if the drift occurred is higher than the previous iteration, then the model is trained repeatedly until the accurate drift is achieved in the research. Fig. 3 shows the drift detection graph.

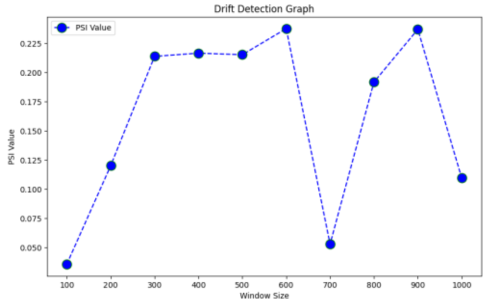


Fig. 3. Drift detection graph.

IV. Results and Discussion

The entire outcomes of the research model UES2-DTM are analyzed and depicted in this section. The complete analysis in this section is performed with the performance metrics such as precision, recall, F1-score, and MSE. Further with the described metrics, the conventional mechanisms are also evaluated shows the proposed UES2-DTM model achieves comparatively high outcomes.

A. Experimental Setup

The experiment is carried out in the PyCharm software of version 2022.2.3 in the system with the configuration of Windows 11 operating system and 16 GB RAM storage. The utilization of the experimental setup supports the research to attain high proficiency specifically when working with the streaming data.

B. Dataset Description

1) IoT-23 Dataset [39]: IoT-23 is a network traffic dataset specifically collected from Internet of Things (IoT) devices. It includes 20 instances of malware-infected traffic from IoT devices and 3 instances of benign IoT device traffic. Benign scenarios network traffic was captured from Philips smart LED lamp, Amazon Echo, and a smart door lock by Somfy. Thus data is captured for analyzing real world network behavior. The upcoming details are collected from the 10000 users. Array(['Benign', 'Okiru', 'PartOfAHorizontalPortScan', 'DDoS', 'C&C', 'C&C0HeartBeat'],dtype=object). The label counts of each of them are Benign – 3024, Okiru – 1670, PartOfAHorizontalPortScan – 4428, DDoS – 858, C&C – 17, C&C0HeartBeat – 3.

C. Performance Assessment

The performance of the UES2-DTM model is analyzed in terms of both K-fold (KF) and training percentage (TP) with metrics such as Accuracy, Precision, recall, and Mean Square Error (MSE). TP 80% and KF 10 are evaluated concerning epochs 100 in this section to depict the efficacy achieved at the model in detail. The Precision achieved at KF 10 in the UES2-DTM model is 97.43%, whereas the recall achieved is 94.62%. Similarly, the F1-score of the proposed model attained 96.01%. In contrast, the proposed model is also verified against the error metrics MSE that obtained 4.583. The performance assessment of the UES2-DTM model concerning KF is depicted in Fig. 4.

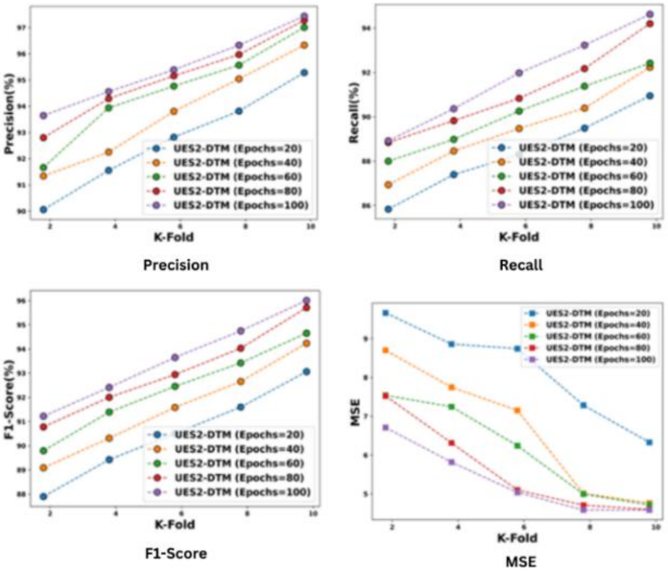


Fig. 4. Performance assessment concerning KF.

The Precision achieved at TP 80% in the UES2-DTM model is 97.19%, whereas the recall achieved is 94.82%. Similarly, the F1-score of the proposed model attained 95.99%. In contrast, the proposed model is also verified against the error metrics MSE that obtained 3.461. The obtained outcomes at both TP and KF analysis are due to efficient mechanisms as well as the models that are combined to detect the anomaly even in the streaming data. The performance assessment of the UES2-DTM model concerning TP is depicted in the Fig. 5.

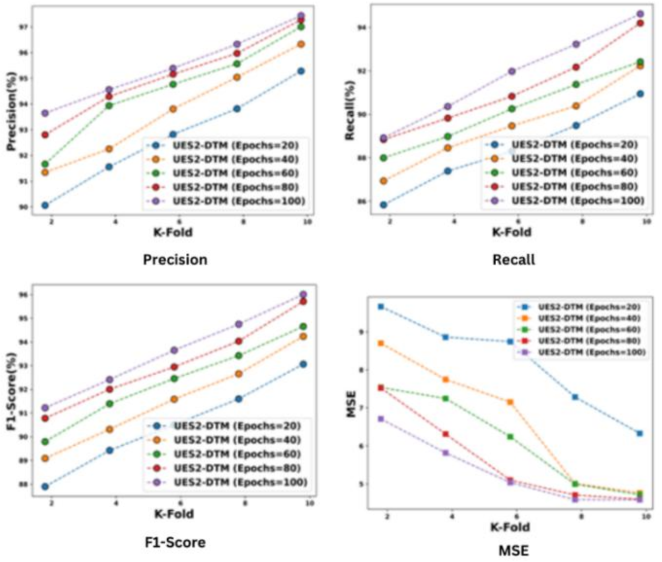


Fig. 5. Performance assessment concerning KF.

D. Comparative Assessment of UES2-DTM Model

The UES2-DTM model is compared with the existing methods such as DNN [28], PiForest [10], RNN [21], SARIMA-BiLSTM [23], DBN-BiLSTM [40], ARO-DBN- BiLSTM [37], and HHO-DBN- BiLSTM [38]. The UES2-DTM model is compared with existing methods concerning the TP in terms of precision achieved at 97.19%, which is improved by 15.44% with DNN, 12.33% with RNN, and 8.72% with DBN-BiLSTM. The recall of the proposed model achieved 94.82% having an average improvement of 15.132% with all the comparative methods. Further, the F1-score of the research model is 95.99%, which shows an improvement of 25.13%, 13.32%, and 8.47% with the respective methods. The MSE obtained in the research model is 3.461, which is an average reduction of 5.66. The comparative assessment concerning TP is illustrated in Fig. 6.

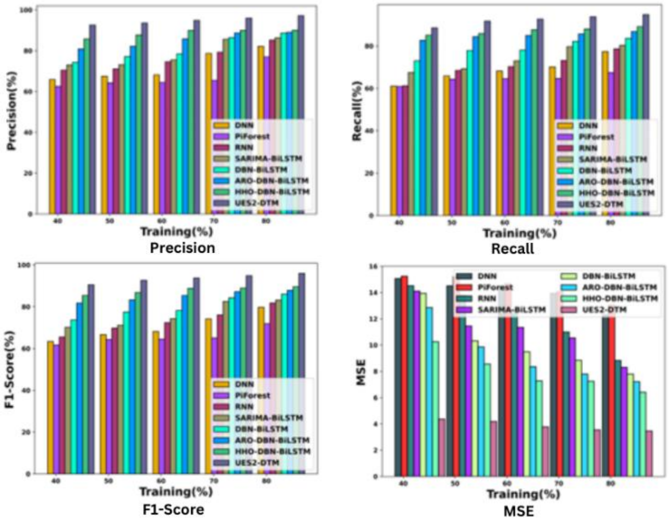


Fig. 6. Comparative assessment concerning TP.

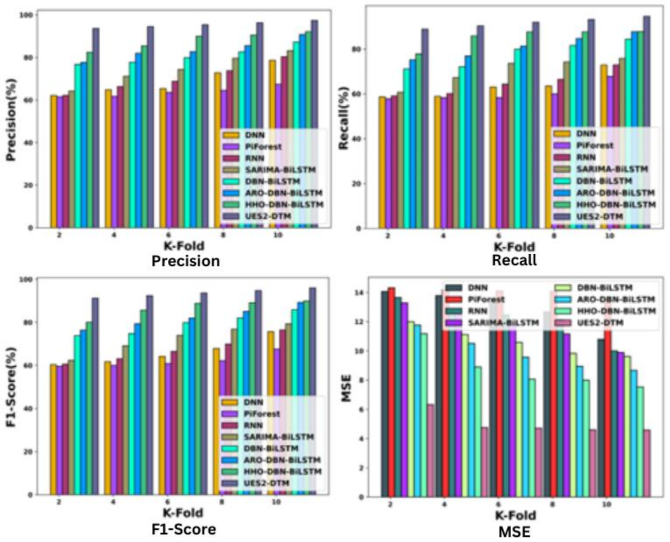
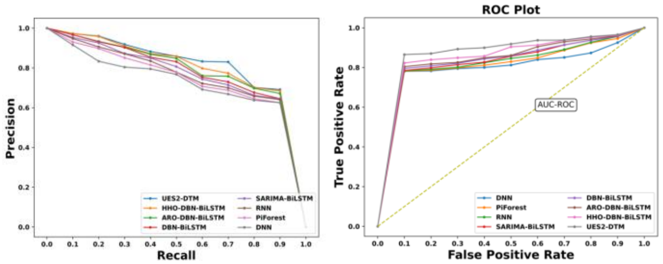


Fig. 7. Comparative assessment concerning KF.

The UES2-DTM model is compared with existing methods concerning the KF in terms of precision achieved 97.43%, which is improved by 19.26% with DNN, 17.48% with RNN, and 10.36% with DBN-BiLSTM. The recall of the proposed model achieved 94.62% having an average improvement of 17.08% with all the comparative methods. Further, the F1-score of the research model is 96.07%, which shows an improvement of 29.58%, 17.33%, and 7.05% with the respective methods. The MSE obtained in the research model is 4.54, which is an average reduction of 5.41. The comparative assessment concerning TP is illustrated in Fig. 7.

E. Graphical Representation of PRC and AUC-ROC

To show the efficiency of the research model, the precision-recall curve (PRC) as well as the area under the receiver operating characteristic curve (AUC-ROC) of the anomaly detection in streaming data is shown in Fig. 8. The PRC represents the interplay between precision and recall, where high precision and high recall indicate low false positive and false negative rates, respectively. The proposed model attained a rate of 0.701 precision, for a sensitivity of 0.8 whereas for 0.9 it obtained a 0.699 rate of positive predicted output. Moreover the AUC-ROC Compares the error rate with the sensitivity rate achieved by the UES2-DTM model. The attained sensitivity of the research model is 0.9653 for an error rate of 0.9. Thus, the proposed research model attains the best outcomes, which is depicted through the evaluation with PRC, and AUC-ROC. The outcomes are due to the enhanced mechanism combinations in the UES2-DTM model.



PRC AUC-ROC

Fig. 8. Graphical representation of PRC and AUC-ROC.

F. Comparative Discussion

The research model is compared with the existing methods that ended up with certain drawbacks in anomaly detection from streaming data.DNN was computationally expensive and required large amounts of data for training, which were further prone to overfitting, especially when dealing with small datasets. PiForest struggled with large-scale streaming data due to its ensemble-based approach. RNNs suffer from gradient vanishing problems during training, and they struggle to capture long-term dependencies in sequences. Combining SARIMA with BiLSTM introduced additional complexity. Choosing the right architecture for DBN-BiLSTM impacted the performance of the research model. The combination of autoencoders, DBNs, and BiLSTM increased model complexity. Thus, the described challenges are overcome with the UES2-DTM model, and the comparative discussion is tabulated in Table II.

TABLE II. COMPARATIVE DISCUSSION OF UES2-DTM MODEL

Analysis / Methods		DNN	Pi Forest	RNN	SARIMA-BiLSTM	DBN-BiLSTM	ARO-DBN-BiLSTM	HHO-DBN-BiLSTM	UES2-DTM
TP=80%	**Precision (%)**	82.18	76.96	85.21	86.32	88.72	89.01	90.07	97.19
	Recall (%)	77.37	67.46	78.62	80.27	83.61	86.84	89.13	94.82
	F1-score (%)	79.71	71.89	81.78	83.19	86.09	87.98	89.63	95.99
	MSE	12.33	13.04	8.83	8.39	7.78	7.21	6.41	3.46
KF=10	**Precision (%)**	78.66	67.44	80.39	83.21	87.33	90.78	92.08	97.43
	Recall (%)	72.88	67.79	72.91	75.85	84.43	87.73	87.79	94.62
	F1-score (%)	75.66	67.61	76.47	79.36	85.86	89.23	89.89	96.01
	MSE	10.79	13.45	10.61	9.88	9.62	8.66	7.52	4.58

V. CONCLUSION

The anomaly detection in the streaming data is performed with the UES2-DTM model that achieves high efficacy in the detection. The research model integrates the UE2O algorithm within an adaptive sliding window framework and adaptive reservoir sampling techniques. By leveraging the UE2O algorithm, this model enhances the accuracy and efficiency of drift detection, ensuring timely and precise identification of anomalies. The adaptive sliding window approach allows for dynamic adjustments to the window size, optimizing the balance between detection sensitivity and computational resource management. Similarly, adaptive reservoir sampling ensures a representative data subset, facilitating effective anomaly detection without overwhelming system resources. Moreover, the involved feature extraction methods significantly augment the model's performance by transforming preprocessed data into meaningful patterns, improving the ability to discern anomalies amidst complex and high-dimensional data. In addition, the preprocessing step not only boosts detection accuracy but also contributes to more robust and interpretable results. Thus, the overall performance of the research model is evaluated with metrics such as precision, recall, F1-score, and MSE that obtained 97.19%, 94.82%, 95.99%, and 3.46 respectively. Future research can include the integration of different DL mechanisms as well as the combination of several advanced optimization mechanisms. In addition, the scalability of the detection model can be evaluated with diverse methods and metrics.

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Comput. Netw., vol. 38, no. 4, pp. 393–422,2002.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.

[3] F. Wang and J. Liu, "Networked wireless sensor data collection: Issues, challenges, and approaches," IEEE Commun. Surv. Tut., vol. 13, no. 4, pp. 673–687, Oct.–Dec. 2011.

[4] Ata A, Khan MA, Abbas S, Ahmad G, Fatima A. Modelling smart road traffic congestion control system using machine learning techniques. Neural Network World. 2019 Mar 1;29(2):99-110.

[5] Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: a survey. IEEE Commun. Surv. Tutor. 16(1), 414–454 (2013)

[6] Rahmani AM, Babaei Z, Souri A. Event-driven IoT architecture for data analysis of reliable healthcare application using complex event processing. Cluster Computing. 2021 Jun: 24:1347-60.

[7] B.Yan and G. Huang, "Supply chain information transmission based on rfid and internet of things," in 2009 ISECS International Colloquium on Computing, Communication, Control, and Management, vol. 4, Aug 2009, pp. 166–169.

[8] L. Xiao and Z. Wang, "Internet of things: A new application for intelligent traffic monitoring system," Journal of networks, vol. 6, no. 6, pp. 887–894, 2011.

[9] Roldán J, Boubeta-Puig J, Martínez JL, Ortiz G. Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks. Expert Systems with Applications. 2020 Jul 1;149:113251.

[10] Jain P, Jain S, Zaïane OR, Srivastava A. Anomaly detection in resource constrained environments with streaming data. IEEE Transactions on Emerging Topics in Computational Intelligence. 2021 Apr 22;6(3):649-59.

[11] O. Etzion and P. Niblett, Event Processing in Action, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2010.

[12] A. Ahmed, H. Arkian, D. Battulga, and et al. Fog computing applications: Taxonomy and requirements. arXiv preprint:1907.11621, 2019.

[13] W. Fengjuan, Z. Xiaoming, and et al. The research on complex event processing method of internet of things. In ICMTMA, pages 12191222. IEEE, 2013.

[14] S. Zhang, H. T. Vo, and et al. Multi-query optimization for complex event processing in sap esp. In ICDE, pages 12131224. IEEE, 2017

[15] Ziehn A. Complex Event Processing for the Internet of Things. fog.;1(3):4.

[16] J. Chen, L. Ramaswamy, D. K. Lowenthal, and et al. Comet: Decentralized complex event detection in mobile delay tolerant networks. In IEEE, pages 131136, 2012.

[17] I. Kolchinsky and A. Schuster. Real-time multi-pattern detection over event streams. In MOD, pages 589606. ACM, 2019.

[18] M. P. Madumal and et al. Adaptive event tree-based hybrid cep computational model for fog computing architecture. In ICTer. IEEE, 2016.

[19] C. Y. Chen, J. H. Fu, T. Sung, P. F. Wang, E. Jou, and M. W. Feng, "Complex event processing for the internet of things and its applications," in 2014 IEEE International Conference on Automation Science and Engineering (CASE), Aug 2014, pp. 1144–1149.

[20] Kashyap, A.A.; Raviraj, S.; Devarakonda, A.; Nayak, K.S.R.; Kv, S.; Bhat, S.J. Traffic flow prediction models—A review of deep learning techniques. Cogent Eng. 2022, 9, 2010510. [CrossRef]

[21] Liu, J., Bai, J., Li, H. and Sun, B., 2021. Improved LSTM-based abnormal stream data detection and correction system for Internet of Things. IEEE Transactions on Industrial Informatics, 18(2), pp.1282-1290.

[22] Yadav, S.; Gulia, P.; Gill, N.S. Flow-MotionNet: A neural network-based video compression architecture. Multimedia. Tools Appl. 2022, 81, 42783–42804. [CrossRef]

[23] Chahal A, Gulia P, Gill NS, Priyadarshini I. A Hybrid Univariate Traffic Congestion Prediction Model for IoT-Enabled Smart City. Information. 2023 Apr 30;14(5):268.

[24] Majumdar S, Subhani MM, Roullier B, Anjum A, Zhu R. Congestion prediction for smart sustainable cities using IoT and machine learning approaches. Sustainable Cities and Society. 2021 Jan 1;64:102500.

[25] Kamble SJ, Kounte MR. Machine learning approach on traffic congestion monitoring system in internet of vehicles. Procedia Computer Science. 2020 Jan 1;171:2235-41.

[26] Bhuyan, Monowar H., Dhruba Kumar Bhattacharyya, and Jugal K. Kalita. "Network anomaly detection: methods, systems and tools." Ieee communications surveys & tutorials 16, no. 1 (2013): 303-336.

[27] Lee, S., Lu, X. and Reijers, H.A., 2022, May. The analysis of online event streams: Predicting the next activity for anomaly detection. In International Conference on Research Challenges in Information Science (pp. 248-264). Cham: Springer International Publishing.

[28] Hassan, A.F., Barakat, S. and Rezk, A., 2022. Towards a deep learning-based outlier detection approach in the context of streaming data. Journal of Big Data, 9(1), p.120.

[29] Xiang, H. and Zhang, X., 2022. Edge computing empowered anomaly detection framework with dynamic insertion and deletion schemes on data streams. World Wide Web, 25(5), pp.2163-2183.

[30] Jain, S., Jain, P. and Srivastava, A., 2021, December. An Efficient Anomaly Detection Approach using Cube Sampling with Streaming Data. In International Conference on Pattern Recognition and Machine Intelligence (pp. 498-505). Cham: Springer International Publishing.

[31] [Maciąg, P.S., Kryszkiewicz, M., Bembenik, R., Lobo, J.L. and Del Ser, J., 2021. Unsupervised anomaly detection in stream data with online evolving spiking neural networks. Neural Networks, 139, pp.118-139.

[32] Fadlil, A., 2022. K Nearest Neighbor imputation performance on missing value data graduate user satisfaction. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 6(4), pp.570-576.

[33] Prasad, P.C. and Beg, A., 2009. Investigating data preprocessing methods for circuit complexity models. Expert Systems with Applications, 36(1), pp.519-526.

[34] Movahedi, F., Coyle, J.L. and Sejdić, E., 2017. Deep belief networks for electroencephalography: A review of recent contributions and future outlooks. IEEE journal of biomedical and health informatics, 22(3), pp.642-652.

[35] Al-Kateb, M., Lee, B.S. and Wang, X.S., 2007, July. Adaptive-size reservoir sampling over data streams. In 19th International Conference on Scientific and Statistical Database Management (SSDBM 2007) (pp. 22-22). IEEE.

[36] Suryawanshi, S., Goswami, A., Patil, P. and Mishra, V., 2023. Adaptive windowing based recurrent neural network for drift adaption in non-stationary environment. Journal of Ambient Intelligence and Humanized Computing, 14(10), pp.14125-14139.

[37] Khalil, A.E., Boghdady, T.A., Alham, M.H. and Ibrahim, D.K., 2023. Enhancing the conventional controllers for load frequency control of isolated microgrids using proposed multi-objective formulation via artificial rabbits optimization algorithm. IEEE Access, 11, pp.3472-3493.

[38] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H., 2019. Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, pp.849-872.

[39] IoT23Dataset:https://www.kaggle.com/datasets/engraqeel/iot23preprocesseddata

[40] Chen, A., Fu, Y., Zheng, X. and Lu, G., 2022. An efficient network behavior anomaly detection using a hybrid DBN-LSTM network. computers & security, 114, p.102600.