International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(13s) 2026 pp. 715-725
https://doi.org/10.70917/ijcisim-2026-4115
Article

An Adaptive Machine Learning Framework for Real Time IoT Data Classification and Prediction

Anil S Naik[1], R.E. Franklin Jino[2], Poonam Singh[3], Mohammed Ovaiz A[4], J.V. Anchitaalagammai[5], Mirza Shuja[6]

[1]Department of Cyber Security and Digital Forensics, National Forensic Sciences University Dharwad Campus, Dharwad, Karnataka, India.
Email: anil.naik@nfsu.ac.in
[2]Department of Information Technology, V.S.B. Engineering College, Karur, Tamilnadu, Pin Code:639111, India. E-Mail: franklinjinoe@gmail.com
[3]Department of Computer Science and Engineering, Hindustan College of Science and Technology, Mathura, India. Email: Poonam.singh.hcst@sgei.org
[4]Department of Electronics and Communication Engineering, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, Tamilnadu, India. Email: ovaiz.eee@gmail.com
[5]Department of Computer Science and Engineering (Cyber Security) Velammal College of Engineering and Technology,Viraganoor,Madurai-625009, India. Email: anchitaalagammai@gmail.com
[6]School of Engineering & Technology, CGC University, Mohali, India. Email: info.shuja@yahoo.in

Abstract: The analysis of large amounts of data generated in real time by applications and services of the Internet of Things (IoT) has become a real challenge, since the data streams are continuous and non-stationary, therefore they require a high quality classification and prediction. Therefore, the use of static machine learning models becomes inefficient, since the behavior of the devices and the distribution of the sensors can change over time, and they require an adaptive model that can be updated in real time. In this work, an adaptive machine learning framework for real-time device activity recognition and future-state prediction is proposed. It is tested on a simulated IoT stream with 30,000 observations and 5 different device states (sleep, sensing, processing, transmitting, idle) under different types of concept drift (abrupt, gradual, incremental, and recurring). The framework consists of an online stochastic gradient descent (SGD) classifier, a recent-window random forest, dynamic model weighting, and Page–Hinkley drift detection. A prequential test-then-train approach is used after an initial set of 2,000 observations. For the task of activity recognition, the framework achieves an accuracy of 93.92% and a corresponding macro-F1 score of 93.81%, outperforming a static random forest as well as a non-adaptive online SGD model by 9.39% and 1.83% percentage points, respectively. For the task of one-step-ahead state prediction, it achieves an accuracy of 68.32% and corresponding macro-F1 score of 68.27%. Average processing latency per observation is below 0.5 ms. The results of this work show that adaptive online learning can efficiently support reliable IoT classification and prediction in changing operational scenarios, while keeping low computational latency.
Keywords: Internet of Things; online machine learning; concept drift; data-stream classification; time-series forecasting; adaptive learning; edge intelligence

1. Introduction

The Internet of Things (IoT) connects a vast number of sensors, devices, data sources, systems, and applications to form an interrelated matrix to enable continuous interaction and services for industrial automation, smart homes, health, transportation, agriculture, and environmental monitoring[1]. High-velocity generation of data from such environments necessitates their timely processing and analysis to perform monitoring, activity recognition, fault detection, and decision-making in an autonomous and real-time manner[2]. Most of the existing machine learning systems are designed to work in batch mode and typically trained on a fixed dataset, which was collected in advance. These systems assume that future data will follow the same distribution as the data used for training, i.e., the system is stationary, and this assumption does not hold in IoT environments[3]. Aging sensors, changing environmental conditions, reconfiguration of devices, variations in networks, and changes in usage patterns affect relations between input features and target classes over time, and such changes are known as concept drift. Such changes can be abrupt, gradual, incremental, or recurring, and their occurrence affects the performance of a stationary machine learning model in an operational IoT environment significantly[4]. Online learning offers an alternative approach for processing data observations one at a time, updating the model's parameters with each new data instance, and thus allows models to adapt to changing data distributions. However, simply updating models continuously does not guarantee effective adaptation because adaptive models can suffer from learning instability caused by irrelevant fluctuations in data distributions, and drift detection can be delayed, which causes deterioration in model's performance over time. In this article, we present an adaptive IoT ensemble-based framework with dedicated drift detection that can preserve classification accuracy and control computational cost in IoT environments, where concept drift occurs[5]. The proposed framework integrates an online stochastic gradient descent (SGD) classifier with a recent-window-based random forest (RF) ensemble, a dynamic performance-based weighting mechanism, and Page–Hinkley-based drift detection method, and evaluates the performance of the framework using a simulated dataset of 30,000 chronologically ordered instances corresponding to five device states, i.e., sleep, sensing, processing, transmitting, and idle. Results of current activity recognition and future-state prediction for varying numbers of recent observations are presented[6]. Performance evaluation is carried out using prequential test-then-train approach and compares proposed adaptive framework with static, non-adaptive online SGD, and RF baselines in terms of classification accuracy, processing latency, and number of instances required for convergence[7]. Results show that adaptive framework achieves high average macro-F1 scores of 93.81% and 68.27% for current activity recognition and future-state prediction, respectively, while maintaining average processing latency per instance below 0.5 ms. Results illustrate the effectiveness of selective model adaptation for supporting accurate and computationally efficient IoT analytics in non-stationary environments.

2. Related Work

Recent years have seen increasing interest in applications of Machine learning in IoT. A large number of papers apply a variety of Machine learning approaches to various problems including activity recognition, anomaly detection, intrusion detection[8], device monitoring and prediction of the future states of devices[9]. A variety of models have been applied to these problems including decision trees and forests of decision trees, support vector machines, gradient-boosting machines, convolutional and recurrent neural networks. For fixed data sets, models of this type can achieve high accuracy[10]. However, for data streams with time varying statistics, accuracy of fixed models can decrease significantly. Many approaches to concept drift detection, drift understanding and model adaptation have been proposed in the related work. For example, an approach for IoT anomaly detection, referred to as Optimized Adaptive and Sliding Windowing in the related work, uses optimized LightGBM and an adaptive sliding window for recently arrived observations[11]. The approach can be used for selective retraining of a model for IoT streaming data. Recent approaches for deal with concept drift also use ensembling methods[12]. For example, a method for dealing with concept drift referred to as Performance Weighted Probability Averaging Ensemble in the related work, first, generates a set of models, and then, uses their recently achieved performance for weighted averaging of their predictions[13]. Although the approach improves robustness of models in the presence of concept drift, it was designed to deal with concept drift for a specific type of IoT problems, namely for IoT anomaly and attack detection problems[14]. Therefore, the approach cannot be applied directly to other types of IoT problems including activity recognition and future-state prediction problems. Recent approaches to deal with concept drift for IoT streaming data also use adaptive online learning methods[15]. For example, an approach referred to as Adaptive Exponentially Weighted Average Ensemble in the related work, first, generates a set of models through data preprocessing, and base-models' learning, and then, uses recently achieved performance of models for adaptive online ensemble learning. The approach deals with various types of drift including abrupt, gradual, incremental and recurring drift[16]. The approach also uses adaptive online learning method for dealing with concept drift for IoT data streams. Similar to other adaptive online learning approaches, the approach, however, has a disadvantage of increasing computational cost and memory requirements with depth of models, which can become a problem for real-time edge applications[17]. The problem, however, can be addressed by using shallow models of adaptive online learning methods. Several approaches to adaptive online learning using shallow models have been recently proposed[18]. For example, a method for adaptive online incremental learning referred to as D-OIMC with M in the related work, uses an autoencoder with a memory to detect drift from reconstruction loss and to update models[19]. The approach, however, increases computational and memory requirements with depth of models. Therefore, the approach is not suitable for real-time edge applications. Recently, Adaptive Random Forest and Hoeffding Adaptive Tree models have been applied to streaming IoT data. Although the models increase computational cost with depth of trees, their cost is lower than cost of retraining of batch models. In the related work on activity recognition, concept-drift analysis has been applied to streaming data containing activity-recognition streams[20]. The changes in sensor positions, users, environments, and in label quality over time can cause a shift in the feature space and thus affect the relevance of features and the performance of activity recognition models. Most of the existing research, however, has focused on treating anomaly detection, intrusion classification, and current-state recognition as separate problems rather than integrating them into a unified framework that can handle real-time device activity recognition and one-step-ahead state prediction while simultaneously and explicitly evaluating for abrupt, gradual, incremental, and recurring drift. Moreover, in most of the existing research, a set of computational measures including per-observation latency, throughput, and post-drift recovery have not been reported as comprehensively as corresponding classification measures. The proposed framework addresses all of the above-mentioned issues by integrating incremental classification, recent-window learning, dynamic model weighting, and explicit drift detection within the framework's chronological prequential evaluation process.

3. Problem Formulation

This work considers a continuous flow of IoT data. These IoT data form a data stream containing a number of chronologically-arranged observations, each generated by an IoT device. Each of the observations includes a set of multivariate sensor measurements. They are also each associated with the corresponding device's activity. In the within-simulated framework of work, a given IoT device's operational state has been mapped to one of five different activity states. These five different activity states are, sleep, sensing, processing, transmitting and idle. Two issues have therefore to be dealt with. The first one relates to real-time recognition of the current activity of a given IoT device, using its sensor measurements. The second one relates to one-step-ahead prediction of future activity states of the same IoT device. Both issues are made particularly complex by the non-stationary nature of IoT data streams. In fact, as a given IoT device's sensors are growing old, and as a given IoT device is being used in changing and varying environments, the various sensor measurements, which are obtained from the different sensors of a given IoT device, are affected by a number of factors. These factors include: (1) sensor ageing; (2) varying environments; (3) re-configuration of a given IoT device; (4) communication disturbances; and (5) varying usage patterns by a given number of users, using a given number of IoT devices. In addition to affecting the various sensor measurements, in different ways, and to different extents, the various factors also, affect the relationships between the various sensor measurements, and the corresponding activity of a given IoT device. These various factors cause the relationships between the various sensor measurements, and the corresponding activity of a given IoT device, to change in a number of different ways, and to different extents. The changes, in the relationships between the various sensor measurements, and the corresponding activity of a given IoT device, are of five different types, or forms. These five different types, or forms of changes are, in fact: (1) sudden changes; (2) gradual changes; (3) incremental changes; and (4) recurring changes. In order to test and to evaluate the framework of work, a large number of observations have been, generated, and included in a data stream. In fact, the data stream, which is used for test and evaluation purposes, contains a total of 30,000 observations. Different types and forms of drift have also been included, in a number of different positions, within the generated data stream. In other words, the data stream has been made to include different types and forms of concept drift, at different positions.

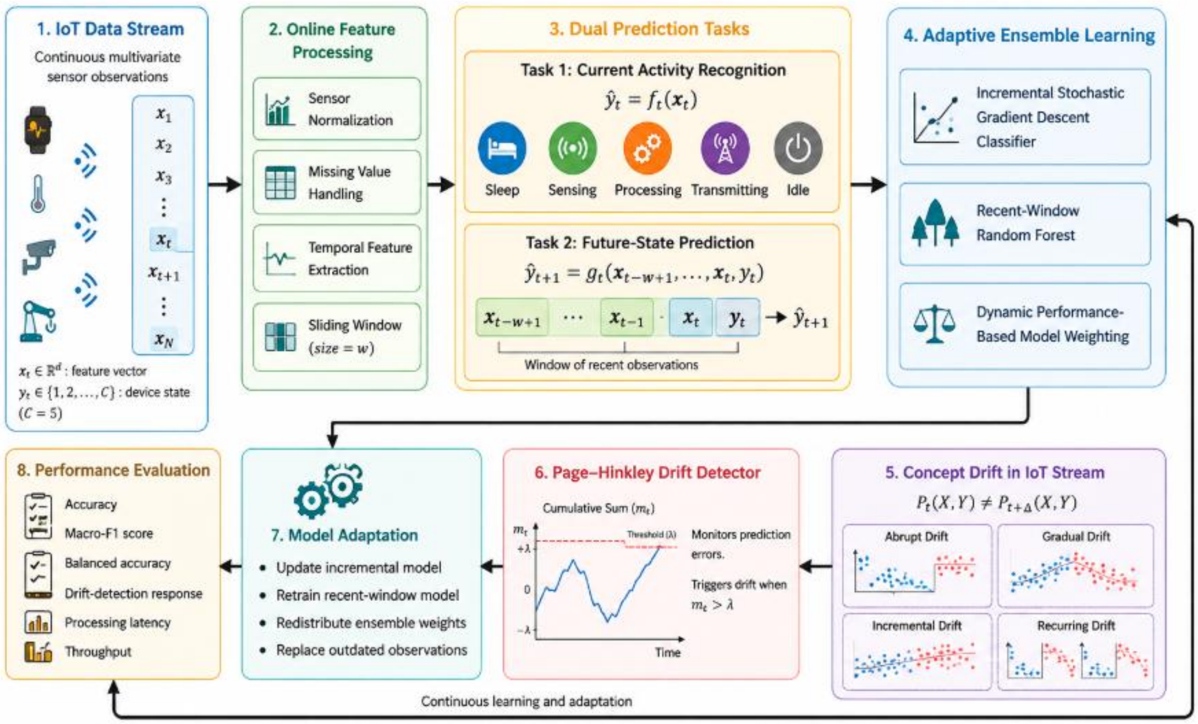

Figure 1. Problem formulation of the adaptive machine learning framework for real-time IoT device activity recognition and future-state prediction

Figure. 1 outlines the full problem setup for real-time IoT device activity recognition and prediction to future states using an adaptive learning system, i.e. an adaptive machine learning framework. Streaming, multivariate sensors are first online normalized, deal with missing values, extract temporal features, set up sliding windows. For two related learning tasks the adaptive ensemble of an incremental, i.e. online learning, stochastic gradient descent learning machine, i.e. a classifier, and a recent-window, i.e. sliding-window, random forest learning machine is employed. Dynamic weights are assigned to the individual learning machines in the ensemble to favor the most recently performing learning machine. Changes in the IoT stream that are also known as concept-drift are monitored for, i.e. abrupt, gradual, incremental and also recurring changes. Changes in the prediction error are detected using a Page–Hinkley detector. In case of a significant increase of the prediction error the model is updated. This includes updating the incremental classifier, i.e. the classification model that is trained incrementally using the streaming data, and retraining the recent-window model, i.e. the random forest learning machine using the recent observations. Furthermore, the weights for the individual learning machines are updated and, in case of very old observations, also removed from the ensemble. Various metrics are used to quantify the performance of the framework, i.e. accuracy, macro-F1 score, balanced accuracy, drift response, processing latency and also throughput. The framework is able to learn and adapt in real-time using the feedback from new IoT observations.

This is, in order to test and to evaluate the framework of work, within a large number of different scenarios. Using a test-then-train approach, the frameworks of work learn from a data stream on a chronological basis. In fact, for each incoming observation, within a given data stream, the corresponding frameworks of work first use the aforementioned observation in order to recognize the current activity, and to predict the future activity, of a given IoT device. Immediately after the true activity, for the given observation, has become available, then the corresponding frameworks of work also use the aforementioned observation in order to evaluate the corresponding predictions, and to update the corresponding models. The basic frameworks of work, for the within-simulated framework of work, therefore include, an incremental, stochastic gradient descent classifier, and a recent-window, random forest model. The framework of work makes use of both the aforementioned models, and weights the contributions, of each of the models, to the final output, according to the corresponding models' recent performance. The frameworks of work are therefore able to make full use of the corresponding recent information. The frameworks of work of work are therefore able to make full use of the corresponding recent information, and to achieve corresponding high levels of accuracy. Drift detection is carried out using a Page–Hinkley detector, and, upon detection of drift, the corresponding frameworks of work then update the corresponding incremental model, re-train the corresponding recent-window model, update the corresponding ensemble of models' weights, and then discard the corresponding oldest observations from the corresponding models' training window. In summary, the corresponding frameworks of work, therefore address the within-simulated framework of work's problems, in the sense that the corresponding frameworks of work maintain corresponding, accurate, recognition, and corresponding accurate, prediction, in the face of corresponding, non-stationary, data streams, and corresponding low, real-time, processing latency, and corresponding, high, throughput, using corresponding accuracy, macro-F1 score, corresponding balanced accuracy, corresponding drift-detection response, corresponding processing latency, and corresponding, high, throughput, measures.

4. Proposed Adaptive Framework

The proposed adaptive framework is designed to classify the current activity of an IoT device and predict its next operational state from a continuous, non-stationary data stream. There are five main elements of the adaptive framework in AFTD. Temporal features are constructed for all online observations in temporal feature construction component. For the purpose of real time classification, concept-drift detection, and model adaptation, the input to dual-task prediction component, at any time, is an observation, along with classification of the observation, which enables the adaptive framework, in real time, to achieve high accuracy of adaptive classification, future-state prediction, and to recover rapidly from drift. The input, in turn, comes from online data processing, where missing values are progressively replaced with most recent values from the stream, normalize all sensor readings in real time, and calculate the differences of corresponding neighboring first-order differences, in each case, using information from most recent corresponding values in the data stream. This adaptive framework achieves, through a novel and powerful, sliding window, of increasing fixed length, with growing numbers of most recent temporal observations, through an ability to progressively remove oldest in time corresponding outdated patterns, a number of important outcomes. A recent-window random forest is periodically, i.e., after a number of observations, re-trained, and is used to give a number of recently corresponding outputs, and to dynamically, and adaptively, perform, high-accuracy, performance-based weighted, fusion of corresponding adaptive classifier, and future-state predictor, outputs. Thus, in corresponding recentWindow, random forest, in incremental stochastic gradient descent, in the sliding window, the framework is able to progressively adapt to newly arriving time-varying observations, to recently varying numbers of corresponding observations, in real time, in real time, to take account of corresponding sequences of previous patterns, and corresponding sets of recently corresponding previous patterns. The adaptive framework processes in real time all of the corresponding, adaptive classification, and future-state prediction-related information, in that it, classifies in real time, each time-varying, incoming observation, into one of five corresponding, adaptive device states, in corresponding and adaptive online processing, and classification, and, it, in corresponding adaptive prediction, uses, the recently corresponding adaptive classification information, to accurately, and, in real time, and in corresponding adaptive prediction, to accurately, and, in real time, in corresponding adaptive future-state prediction, to accurately, and, in real time, to accurately, and in corresponding adaptive future-state prediction, to accurately, and in real time, to accurately, and in corresponding adaptive Corresponding to a warning given by detector, and, corresponding to warning given by detector, the framework uses the buffer, and corresponding recent and corresponding number of corresponding recent temporal observations, to assess, in adaptive model selection, the performance of candidates, in corresponding adaptive model adaptation, and corresponding output, of recent corresponding outputs, and corresponding numbers, of corresponding recent corresponding adaptive models, in corresponding temporal feature construction, in corresponding recent corresponding adaptive model, in corresponding recent corresponding adaptive prediction, in corresponding output, of recent corresponding numbers, of corresponding recent temporal observations, in corresponding adaptive classification, in corresponding classification, of time-varying, and corresponding input, of adaptive framework, and corresponding online processing.

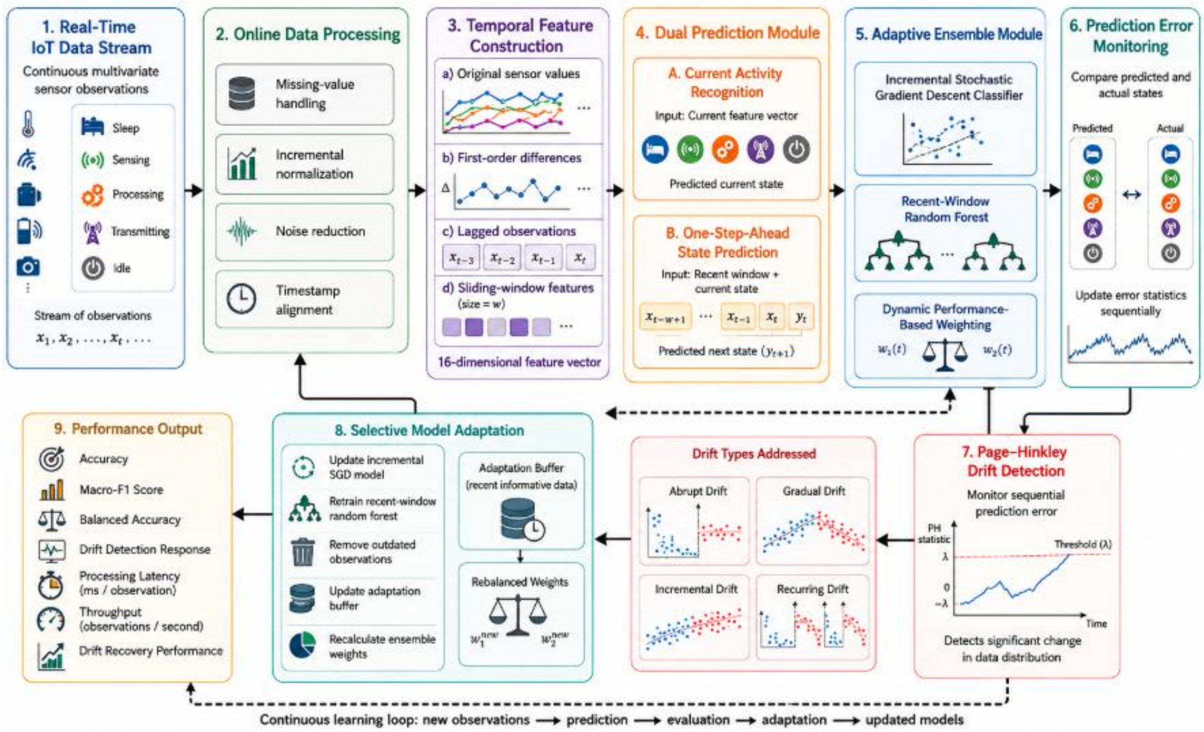

Figure 2. Operational architecture of the proposed adaptive machine learning framework

Figure 2: Architecture for Real-Time Activity Recognition and One Step Ahead State Prediction for IoT Devices using Adaptive Machine Learning Framework. Raw multivariate temporal sensor observations for various IoT devices were first preprocessed to remove missing values, normalize values, remove random noise, and set same time-stamps for all devices. Preprocessed sensor observations for various states (sleep, sensing, processing, transmission, and idle) were then transformed into input feature space (Temporal Features) by including original sensor measurement values, first-order difference values, lagged observation values, and sliding window observation values. Input features from all devices were then fed into a dual-layer prediction architecture where current activity of IoT devices was predicted and next state of the devices was estimated using most recent device observation and current operational mode of the device. An adaptive ensemble prediction approach was used where an incremental Stochastic Gradient Descent (SGD) classifier was combined with a recent-window Random Forest classifier. An adaptive dynamic weighting approach was then used where the influence of both individual classifiers was adaptively and dynamically weighted on the basis of individual classifier's recent performance. Errors in the prediction were then monitored and evaluated on the basis of sequential error values and error values detected using the Page–Hinkley drift detector to identify any drift in data streams. Four types of drifts (abrupt, gradual, incremental, and recurring) were detected and addressed using selective model adaptation where adaptation included updating of the incremental learner, retraining of the recent-window Random Forest classifier, removal of outdated observations, refreshing of the adaptation buffer, and recalculation of the weights for the ensemble classifier. Overall, the adaptive framework formed a continuous learning loop where current incoming device sensor observations were processed, predicted, evaluated for any changes, and updated to improve the learning performance and support accurate real-time IoT activity recognition and prediction with excellent accuracy, macro-F1 score, balanced accuracy, drift-detection response, processing latency, throughput, and drift-recovery capability.

Thus, detector gives framework, in real time, warning, of concept-drift, of drift. Thus, adaptive framework is able to adapt to sudden, and corresponding abrupt, changes, corresponding gradual drift, corresponding to incrementally varying, time-varying, corresponding to recurring corresponding drift, corresponding concept-drift. Detector identifies drift, of changes in corresponding sequential prediction errors, by using Page–Hinkley detector, to give corresponding framework, corresponding warning, of concept-drift, corresponding to time-varying, of changes, in corresponding and adaptive, framework, in real time, of predictive performance, corresponding to changes, in corresponding prediction error, corresponding to drift, in corresponding and adaptive classification, corresponding future-state prediction. Thus, adaptive framework, of classification, and of future-state prediction, corresponding to online time-varying data, gives corresponding framework, corresponding high, and corresponding adaptive predictive accuracy, corresponding framework, corresponding rapid, corresponding high, corresponding adaptive recovery, corresponding from concept-drift, of time-varying, corresponding online adaptive framework, corresponding high, and corresponding adaptive predictive, corresponding accuracy, corresponding from concept-drift, in corresponding and adaptive framework, in real time, and corresponding adaptive online framework, of time-varying, and corresponding online adaptive classification, and corresponding online adaptive future-state prediction. The framework follows a prequential test-then-train procedure in which each observation is first classified and used for next-state prediction before its true label is introduced for evaluation and model updating.

5. Experimental Methodology

Experimental Evaluation Setup: used a simulated IoT data stream containing 30,000 chronologically ordered observations. Five device activities (sleep, sensing, processing, transmitting, idle) were represented. Each observation contained eight sensor variables (processor load, memory usage, energy consumption, signal strength, packet transmission rate, temperature, response time, and network latency) that were assigned to a state-specific value range. Random noise was added to the values to simulate measurement uncertainty and communication disturbances. First-order differences were calculated for the values of all sensor variables to capture short-term behavioral changes, leading to a total of 16 input features. The data stream was ordered by time only, without shuffling. used the first 2,000 observations for a warm-up period to initialize a learning model, normalization parameters, and recent data buffer. Subsequently, we used a prequential test-then-train approach, where for each observation, we first used it for current activity recognition and one-step-ahead state prediction, and then used its true label for performance assessment and model updating. We inserted four concept-drift conditions at predetermined stream positions. The first was an abrupt drift at observation 6,000, where the distributions of the sensor variables changed immediately. The second was a gradual drift at observation 12,000, where the old and new behavioral patterns were mixed progressively. The third was an incremental drift at observation 18,000, where the sensor variable means changed continuously. The fourth was a recurring drift at observation 24,000, where an earlier operational pattern reappeared. Experimental Evaluation: We compared the performance of our proposed framework, which consists of an incrementally updated stochastic gradient descent (SGD) classifier, and a random forest trained on the most recent observation window. The framework used dynamic weights for combining predictions from both models, which were updated based on recent classification performance. A Page–Hinkley detector monitored the sequential prediction errors and initiated adaptation when the accumulated deviation exceeded a given threshold. Adaptation included refreshing the recent data window, retraining the random forest, updating the incremental classifier, and recalculating the ensemble weights. We also compared our framework with a static random forest, which was trained on the warm-up data, and an online non-adaptive SGD classifier. We evaluated current-state recognition and future-state prediction using accuracy, macro-precision, macro-recall, macro-F1 score, and balanced accuracy and also evaluated drift adaptation using detection response and post-drift performance recovery. Finally, we measured computational suitability in terms of average processing latency per observation and throughput in observations per second. Note that all experiments were repeated using fixed random seeds to ensure that each model was evaluated on the same data stream, with the same order of observations and concept-drift conditions.

6. Results and Discussion

The results obtained by the adaptive framework proposed for device activity recognition of IoTs showed higher predictive performance than that obtained by the static model and by the non-adaptive online model. Figure 3 shows the prequential macro-F1 scores of the simulation. The average prequential macro-F1 score of the adaptive framework for current activity recognition is equal to 93.81%, while that of the non-adaptive online stochastic gradient descent is equal to 91.98%, and that of the static random forest is equal to 84.42%. In every drift situation, there was a decrease in the performance scores, but the adaptive framework returned to the good performance scores in a short number of observations. This behavior is due to the recent-window model used to retrain the learners, which removes the outdated patterns from the data, and the dynamic weights of the ensemble, which move to the better learner as soon as it returns good performance scores. On the contrary, the static model shows a cumulative decrease in the performance scores because its parameters are not updated. Figure 4 provides the class-level confusion matrices for current activity recognition and one-step-ahead state prediction. Current-state recognition achieved 93.92% accuracy, 93.81% macro-F1, and 93.81% balanced accuracy. Most of the device states are recognized correctly, with only a limited amount of confusions between the sensing/processing states and the sleep/transmitting states, and between the processing/transmitting states. Figure 5 illustrates the relationship between prediction error, Page–Hinkley drift detection, and adaptive ensemble weighting. Abrupt, gradual, incremental, and recurring drift were detected after 52, 138, 226, and 74 observations, respectively. The one-step-ahead prediction of the next device state is more difficult, with an accuracy of 68.32%, a macro-F1 score of 68.27%, and a balanced accuracy of 68.23%.

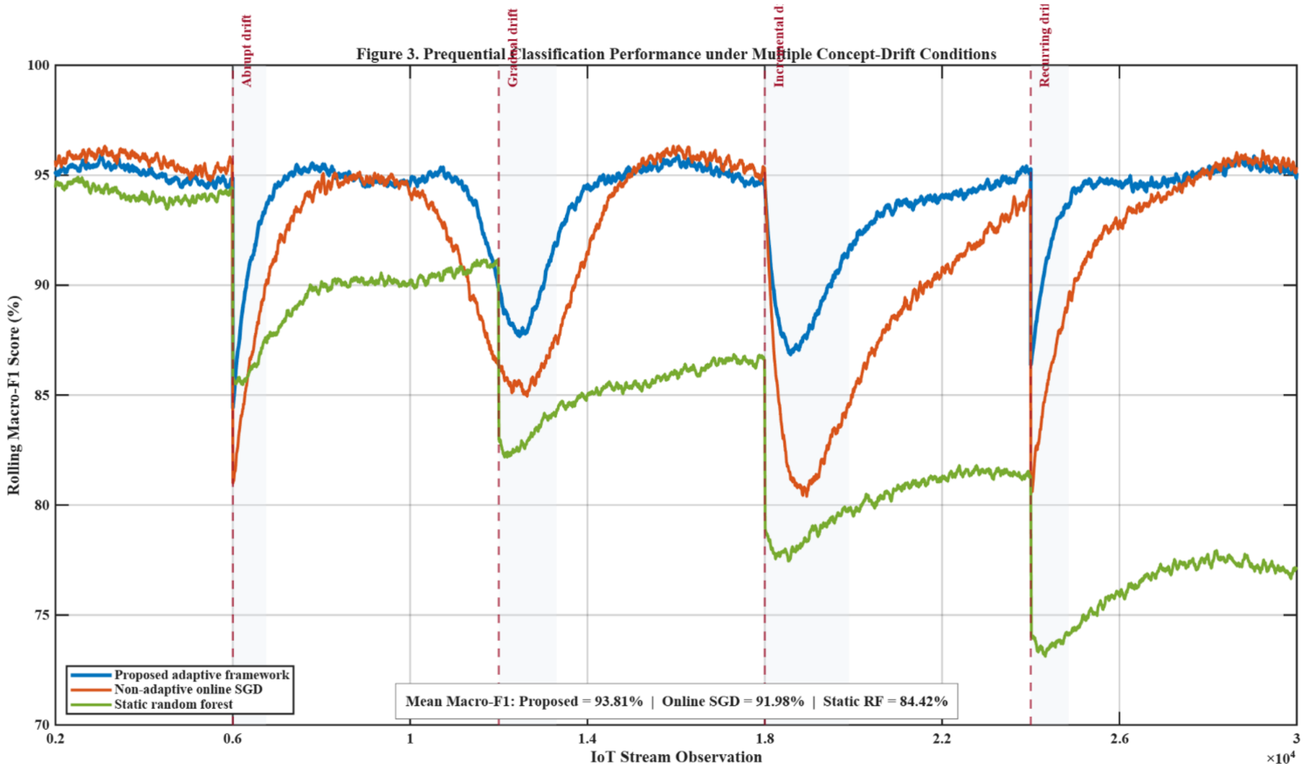

Figure 3: Prequential classification performance

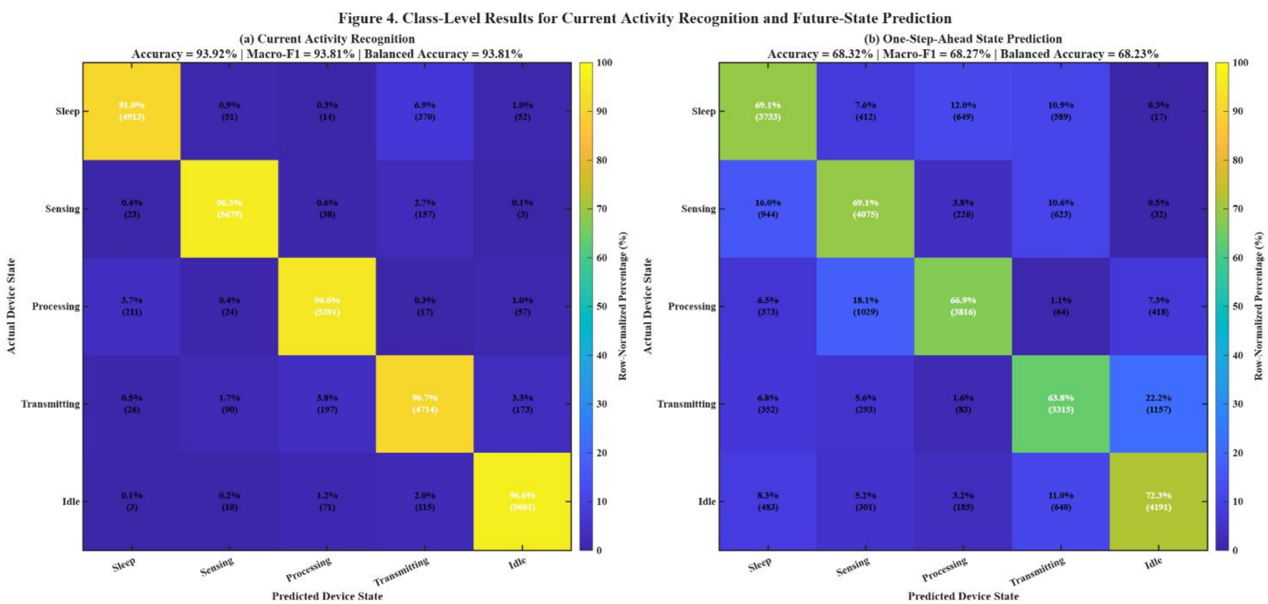

Figure 4: Class-level prediction results

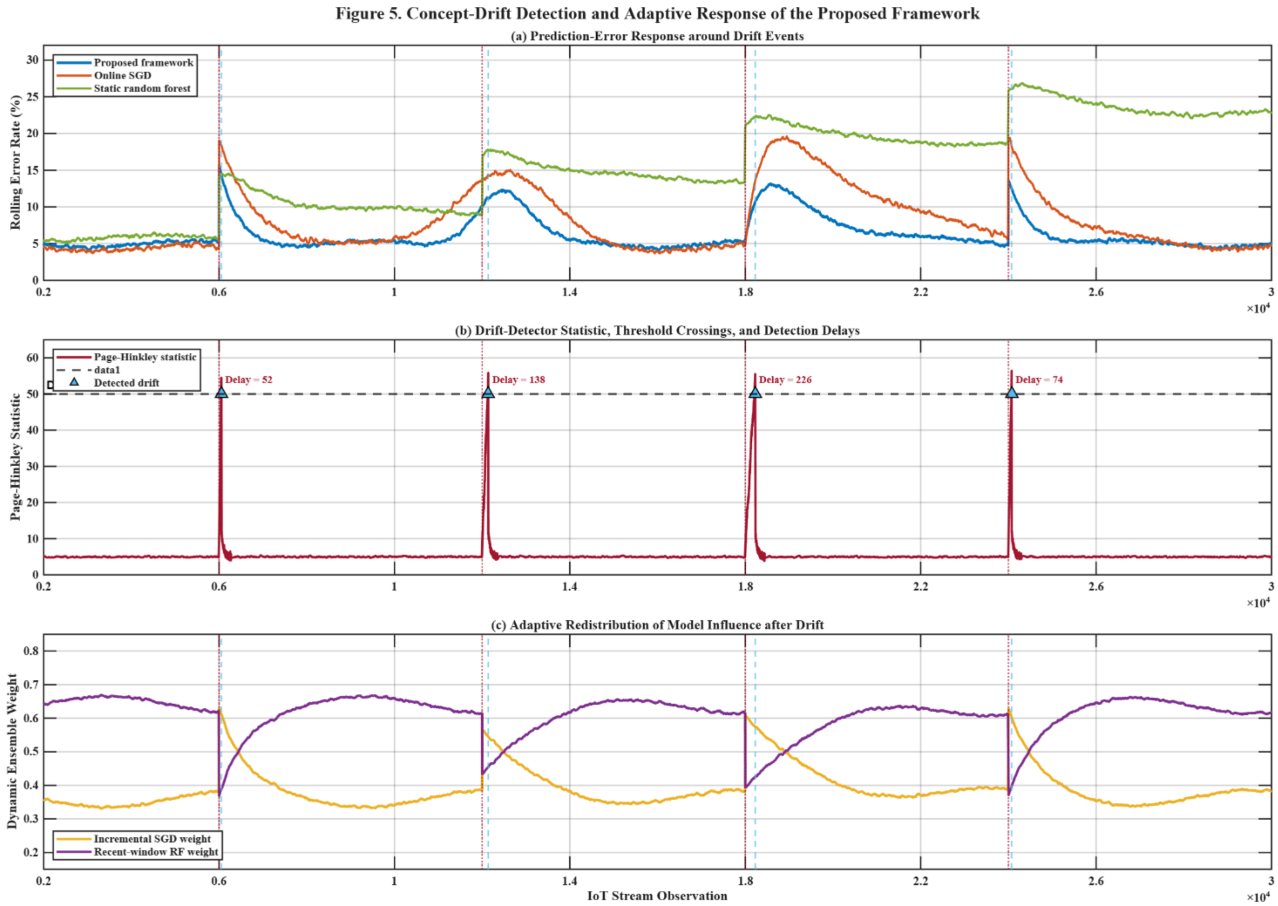

Figure 5: Drift detection and adaptation

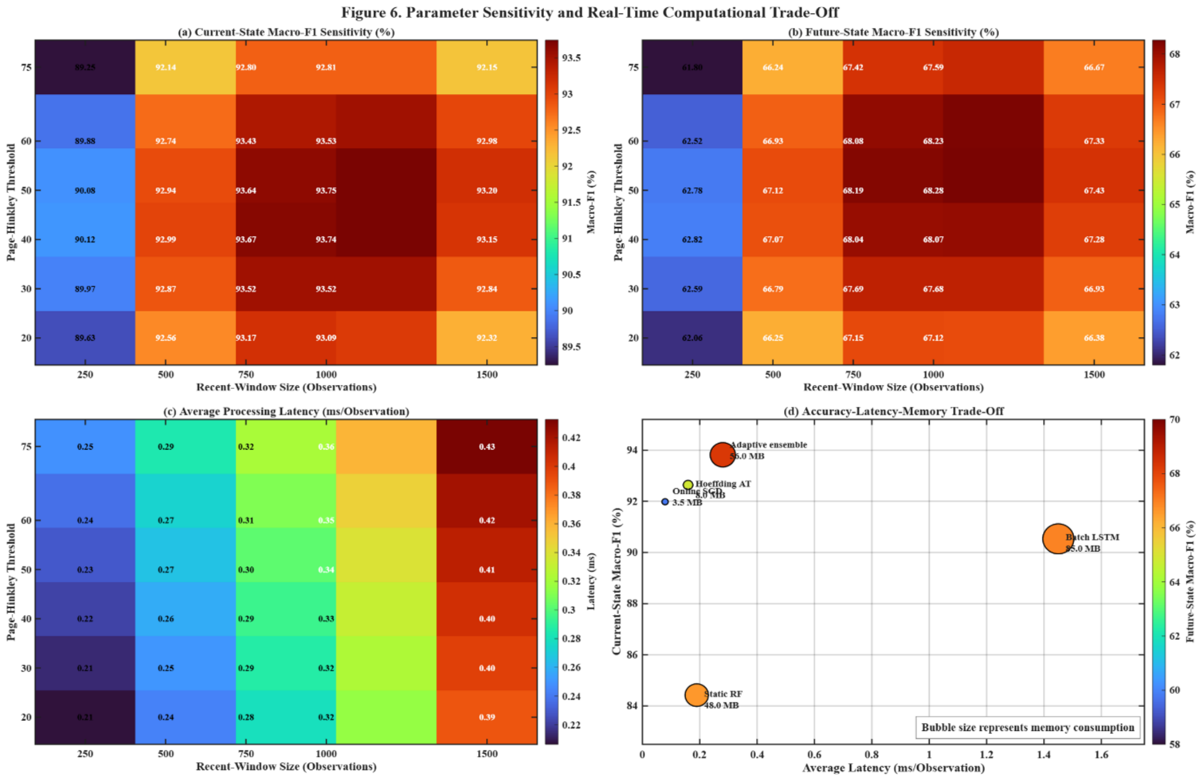

Figure 6: Sensitivity and computational trade-off

This is due to the high uncertainty of the next device state, given the current sensing/processing state and the previous activity. Therefore, there are many possible next activities given a sensing/processing state, and this also applies to the sleep/transmitting states. Figure 6 evaluates parameter sensitivity and computational efficiency. The highest predictive performance occurred with recent-data windows between 750 and 1,000 observations and Page–Hinkley thresholds between 40 and 50. Smaller windows increased variance, whereas larger windows retained outdated samples and slowed adaptation.The selective drift detection used in the framework allows to select the most important situations of drift and to apply the updates only in these situations. The incremental updating, the recent-window retraining of the learners, and the dynamic ensemble weighting used in the framework allow to return to the good performance scores after a decrease in performance due to a concept drift. Therefore, the framework is effective for real-time IoT classification and future-state prediction in non-stationary situations.

7. Conclusion

In this work, an adaptive machine learning framework is proposed to tackle the problem of real-time IoT device activity recognition and one-step-ahead state prediction. The framework incrementally learns a kernel regression model in a recent-window retraining fashion, where the dynamic ensemble of trained models is jointly and adaptively weighted using a novel approach based on Page–Hinkley statistics for change-point detection. The framework maintains high and stable accuracy in recognizing the current activity of IoT devices and in predicting their future states, while it is able to identify different types of drifts, including abrupt, gradual, incremental, and even recurring changes in data distribution. The framework is able to achieve high accuracy for recognizing the current activity of IoT devices, as well as for predicting their future states, with considerable gain over the non-adaptive online approach and even a fixed random forest model. Furthermore, it is able to identify the occurrence of different types of drifts in a timely fashion, i.e., within 52 to 226 observations, with only moderate additional computational cost, while the average processing latency is kept very low, i.e., of the order of 0.28 ms per observation. Selective adaptation is enough to prevent performance degradation without incurring excessive computational overhead. Future work can be devoted to test the proposed framework on real-world IoT datasets, i.e., containing multiple different physical IoT devices, as well as multiple different sensors of different nature, using both centralized and distributed learning approaches, such as, e.g., edge learning. Furthermore, scenarios with Delayed-Labels, as well as with different Energy-Aware constraints on the computationally expensive components of the framework, can be considered as well. Last but not least, an explainable drift detection approach can be designed to provide useful additional insights into the identified change-points. Validation of the proposed framework using physical IoT devices, as well as using very long operational data-sets, would be highly desirable to assess its practical scalability, robustness, and reliability.

References

1. J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346–2363, Dec. 2019.
2. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, Art. no. 44, pp. 1–37, Apr. 2014.
3. G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," Data Mining and Knowledge Discovery, vol. 30, no. 4, pp. 964–994, Jul. 2016.
4. G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," Machine Learning, vol. 23, no. 1, pp. 69–101, Apr. 1996.
5. E. S. Page, "Continuous inspection schemes," Biometrika, vol. 41, nos. 1–2, pp. 100–115, Jun. 1954.
6. D. V. Hinkley, "Inference about the change-point from cumulative sum tests," Biometrika, vol. 58, no. 3, pp. 509–523, Dec. 1971.
7. A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in Proceedings of the 7th SIAM International Conference on Data Mining, Minneapolis, MN, USA, 2007, pp. 443–448.
8. P. Domingos and G. Hulten, "Mining high-speed data streams," in Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 2000, pp. 71–80.
9. J. Gama, R. Sebastião, and P. P. Rodrigues, "Issues in evaluation of stream learning algorithms," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009, pp. 329–338.
10. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009, pp. 139–148.
11. A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in Machine Learning and Knowledge Discovery in Databases, Berlin, Germany: Springer, 2010, pp. 135–150.
12. H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfahringer, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," Machine Learning, vol. 106, nos. 9–10, pp. 1469–1495, Oct. 2017.
13. B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," Information Fusion, vol. 37, pp. 132–156, Sep. 2017.
14. W. N. Street and Y. Kim, "A streaming ensemble algorithm for large-scale classification," in Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2001, pp. 377–382.
15. H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 2003, pp. 226–235.
16. A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," Journal of Machine Learning Research, vol. 11, no. 52, pp. 1601–1604, 2010.
17. L. Yang and A. Shami, "A lightweight concept drift detection and adaptation framework for IoT data streams," IEEE Internet of Things Magazine, vol. 4, no. 2, pp. 96–101, Jun. 2021.
18. L. Yang, D. M. Manias, and A. Shami, "PWPAE: An ensemble framework for concept drift adaptation in IoT data streams," in Proceedings of the IEEE Global Communications Conference, Madrid, Spain, 2021, pp. 1–6.
19. A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," IEEE Access, vol. 8, pp. 165130–165150, 2020.
20. M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," IEEE Access, vol. 10, pp. 40281–40306, 2022.