EXPLORING MACHINE LEARNING ALGORITHM PERFORMANCE ACROSS VARIED IOT DATA DOMAINS

Amit Chaurasiya[1], Dr. Dhananjay Bisen[2]
1 Department of Center of Internet of Things, Madhav Institute of Technology and Science, Gwalior 474005, MP, India
2 Department of Center of Internet of Things, Madhav Institute of Technology and Science, Gwalior 474005, MP, India

Abstract

The past few years have witnessed a remarkable change by the rapid expansion of the Internet of Things (IoT). The increasing spread of networked devices in various areas of the Internet of Things has resulted in a huge amount of machine data. This data contains rich information that can be harnessed across domains such as healthcare, agriculture, smart cities, and manufacturing. To effectively analyze and capitalize on this information, the integration of artificial intelligence (AI) and machine learning (ML) methods is essential.

We assess the effectiveness of six well-established machine learning algorithms for classification on three datasets. The algorithms are pitted against each other using metrics like precision, recall, F1-score, accuracy, and execution speed. In a comprehensive evaluation encompassing precision, recall, F1-score, accuracy, and execution time, Random Forests consistently outperformed other machine learning models. This research highlights the potential for optimizing data analytics in IoT applications through the careful selection and implementation of effective ML algorithms.

Keywords: Internet of Things, IoT Datasets, Dataset analysis, ML, Machine learning Performance comparison.

1. Introduction

The Internet of Things [1] is an upcoming technology for the future. Its application areas are endless and the research potential is enormous. Information generation has provided decision makers with a wealth of data. Extracting value and knowledge from these rapidly growing datasets necessitates research into efficient processing solutions. The European Research Group on the Internet of Things (IERC) defines IoT as "a dynamic and self-organizing global network infrastructure based on standard communication and communication protocols, that there are physical and virtual 'things' A virtual person using materials and intelligent interfaces and seamlessly integrates into communication networks" [2].

IoT refers to a system in which real-world "things" and sensors that are on or near these things are connected to the Internet via a wired or wireless network infrastructure. With IoT technology, the world will be elegant in every way [12]. As the IoT is added to many important applications, it offers opportunities for smart cities, smart healthcare, smart homes and buildings.

To address these challenges, leveraging artificial intelligence (AI), particularly machine learning (ML), is a promising approach [4]. Machine Learning tackles the challenge of data analysis by offering a powerful arsenal of algorithms. These algorithms can sift through data, uncover hidden patterns and relationships, and even predict future trends based on past information. These capabilities can enhance the intelligence of IoT systems and optimize their decision-making processes.

Machine learning automates the process of building analytical models from data, enabling computers to learn and improve without constant human intervention. Machine learning, a field within artificial intelligence, empowers systems to learn from data, identify patterns, and make autonomous decisions [13]. Machine learning is an important component of the growing field of data science. New computer technologies make today's machine learning different than it used to be. Machine learning's strength lies in its ability of models to continuously learn and adapt to new data, without human intervention.

With the continuous development of IoT and the introduction of ML, it is important to evaluate the role of machine learning in the context of IoT data. Although many researchers have investigated the application of ML to specific IoT problems, a comprehensive evaluation in multiple domains is still needed. To address this knowledge gap, this study critically evaluates the performance of six popular classification algorithms on three IoT datasets. [15] The evaluation compares these algorithms using various metrics such as precision, recall, F1 score, and execution time.

The comprehensive experiments showed that Random Forests outperformed other ML models in all performance measures. By providing insight into the performance dynamics of these algorithms, this study will help stakeholders find suitable algorithms for specific IoT applications and advance the field of IoT data analytics.

2. Literature Review

This section delves into recent research that leverages machine learning (ML) for diverse applications within the Internet of Things (IoT) domain. Our focus is on studies that analyze the performance of various ML algorithms for specific IoT-related tasks.

Several studies have investigated the efficacy of ML algorithms on various IoT datasets. [6] compared five supervised learning algorithms across five different datasets. Interestingly, Decision Trees consistently achieved the highest accuracy (99%), while Logistic Regression and Naive Bayes lagged behind. This suggests that the choice of algorithm can significantly impact performance, and simpler models may not always be inferior.

Research by [7] explored hyperparameter optimization for a Support Vector Regression (SVR) model used to predict air-conditioning load in a shopping mall. Their findings highlight the potential benefits of fine-tuning models for specific tasks. They demonstrated that hybrid models like Chaos-SVR and WD-SVR can outperform single models, although at the cost of increased complexity.

The study by [8] compared various classification algorithms for an activity recognition problem using android phone accelerometer data. Notably, IB1 and IBk (lazy learners) emerged as the most accurate for hand palm position detection. This underscores the importance of considering the nature of the problem and the data characteristics when choosing algorithms.

Ever et.al [9] investigated the influence of dataset size and features on the performance of four ML algorithms for prediction tasks. They concluded that the number of data samples has a limited impact, but the specific features within the dataset significantly affect performance. Additionally, their findings suggest that neural network models often deliver superior and better results.

[10] This study investigated a new machine learning method called the Conditional Restricted Boltzmann Machine (CRBM) for predicting energy use in office buildings. The results showed that CRBMs achieved superior performance compared to traditional approaches like Artificial Neural Networks and Hidden Markov Models for time series forecasting tasks in the Internet of Things (IoT) domain.

[11] Evaluated the effectiveness of various machine learning algorithms in forecasting human activities within a smart home setting. Their research revealed that Support Vector Machines achieved the best accuracy, particularly when the network utilized data from all available sensors. This emphasizes the potential benefits of sensor fusion in enhancing the accuracy of ML-based action recognition systems.

By examining these studies, we gain valuable insights into the performance of various ML algorithms for diverse IoT applications. We observe the importance of considering factors like dataset characteristics, task-specific optimization, and sensor fusion strategies when selecting and implementing ML solutions within the ever-evolving world of IoT.

3. Proposed Methodology:

The methodology for this research involves evaluating the performance of six popular machine algorithms in the context of IoT data analysis. The proposed approach comprises the following steps:

3.1.1 Data Selection

Utilize three IoT-related datasets from diverse domains such as healthcare, smart cities, and agriculture. These datasets should vary in size, features, and complexity to provide a comprehensive evaluation.

3.1.2 Algorithm Selection:

Choose six widely-used machine and deep learning algorithms, including Random Forests, Support Vector Machines, Decision Trees, Neural Networks, and others, based on their relevance and effectiveness in IoT data analysis.

3.1.3 Preprocessing:

Perform data preprocessing, including data cleaning, normalization, and transformation, to ensure high-quality inputs for the algorithms.

3.1.4 Model Training:

Train each of the selected algorithms on the IoT datasets using appropriate hyperparameter tuning techniques to optimize their performance.

3.1.5 Performance Evaluation:

Evaluate the trained models using key performance metrics such as precision, recall, F1-score, accuracy, and execution time. Conduct these evaluations across all datasets to provide a comprehensive comparison.

3.1.5 Analysis and Interpretation:

Analyze the results from the performance evaluations to identify the strengths and weaknesses of each algorithm in different IoT contexts. Identify which algorithms are most effective for specific IoT domains and data types.

3.1.6 Recommendations:

Based on the findings, provide recommendations on the best-performing algorithms for different IoT applications and highlight potential areas for future research and improvement.

4. Datasets

The type of data plays a critical role in selecting the most effective machine learning algorithms. Existing research on ML algorithm performance often focuses on specific problems and may not generalize well to the complexities of IoT data. A key strength of this paper is our utilization of diverse IoT datasets from distinct domains, each with unique characteristics. These datasets originate from real-world "smart environments" and were collected using actual IoT devices.

To showcase this data diversity, we leverage three distinct datasets, referred to as Dataset 1, Dataset 2, and Dataset 3 for clarity.

- **Dataset 1** [5]: The dataset in question is an Elderly Fall Prediction and Detection dataset, designed by 'cStick' with the purpose of assisting both visually and hearing-impaired older adults. The 'cStick' system is capable of not only detecting falls but also predicting the likelihood of falls to help prevent their occurrence. It can monitor the environment, alert the user to previous fall incidents at specific locations, and provide updates on the location and its surroundings. The dataset provided is in CSV format and includes several parameters: Distance, minimum, average, and maximum pressure readings, along with heart rate variability (HRV), blood sugar levels, and accelerometer data (categorized as above or below a specific threshold) were analyzed to investigate their relationship with fall types (no fall, slip/trip/fall prediction, or definite fall).
- **Dataset 2** [19]: Our research utilizes a comprehensive Australian weather dataset encompassing daily observations from various stations. Sensors capture temperature, rainfall, sunshine, wind, humidity, pressure, and cloud cover. This data empowers us to train machine learning models for a key task: predicting the likelihood of rain tomorrow, impacting applications from agriculture to infrastructure planning.
- **Dataset 3** [20]: This dataset provides a unique opportunity to explore sensor fusion to predict door failures. A dataset designed for predictive maintenance of elevator doors. Real-world sensor readings (4Hz) are collected during peak and evening building usage (16:30-23:30). The dataset project aims to utilize sensor data to minimize unplanned elevator shutdowns during peak usage hours (4:30 PM to 11:30 PM). By predicting the absolute value of vibration, the system can identify potential door malfunctions early on, enabling proactive maintenance and preventing disruptions. The dataset has 6 columns and 1,12,001 rows.

5. Performance Evaluation Metrics

To evaluate the effectiveness of machine learning algorithms, we use various tools called performance evaluation metrics. These metrics help assess how well the algorithms have performed their tasks by focusing on different aspects of performance. Choosing the right set of metrics for each machine learning problem is essential. This paper uses several common metrics for classification tasks to gain insights into algorithm performance and conduct a comparative analysis. These metrics include precision, recall, F1-score, accuracy, confusion matrix, and ROC-AUC score.

- **Precision:** Precision measures how many of the data items an algorithm has predicted as positive are actually relevant. In other words, it calculates the proportion of true positive predictions out of the total positive predictions made by the algorithm.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

- **Recall:** Recall indicates how many of the relevant data items have been correctly selected by the algorithm. It is the proportion of true positive predictions out of the total actual positive cases.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ negatives} \quad (2)$$

- **F1-score:** the F1-score combines precision and recall into a single metric. This F1-score, also called F-measure, acts as a harmonic mean, offering a balanced assessment.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

- **Accuracy:** A common way to measure how well a classification model performs is by looking at its accuracy. It measures the ratio of correctly classified data items to the total number of observations. However, it may not always be the best metric, especially when the dataset has unbalanced classes.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

6. Experiment and Results

This section delves into a comparative analysis of various machine learning algorithms (referenced in [14]) on the three datasets introduced earlier. We assess their performance using a comprehensive set of metrics: precision, recall, F1-score, precision on both training and testing data, and execution time.

For each dataset, individual tables illustrate the comparative performance of the applied algorithms. It's important to note that the Support Vector Machine (SVM) was excluded from the analysis of dataset 2 due to exceeding a predefined time limit. Within the tables, the highest values for each metric are bolded for clear comparison.

Certain models showed high accuracy at the cost of considerable execution time. However, for resource-constrained IoT devices, minimizing processing time is critical to maintain system responsiveness and energy efficiency. Therefore, in the developed models, the first place is the balance between achieving good results and effective execution.

To ensure consistency across all experiments, a dedicated machine served as the testing ground. This workhorse boasted an 11th generation Intel Core i5 processor humming at 2.4 GHz, 8 GB of RAM to keep things running smoothly, and a 512 GB solid-state drive for swift data access. Windows 11 provided the operating system foundation, while Python 3.7 acted as the programming language of choice. For model development, Jupyter Notebook offered a user-friendly interface. To empower these models, a powerful arsenal of machine learning libraries was employed, including scikit-learn, Pandas, and time.

6.1 Datasets Result:

1) Dataset 1: [5] This dataset provided is in CSV format and includes several parameters: Distance, minimum, average, and maximum pressure readings, along with heart rate variability (HRV), blood sugar levels, and accelerometer data. The performance comparison of various machine learning algorithms revealed that Random Forest exhibited the highest precision, recall, and F1-

score at 98%, along with a perfect training accuracy of 100% and a test accuracy of 98/99%. Decision Tree and K-NN also demonstrated high accuracy and low execution times, with K-NN having the shortest execution time of 0.06 seconds. On the other hand, SVM, despite its high precision (96%) and recall (97%), had the longest execution time at 2.70 seconds, indicating its computational intensity. Logistic Regression and Naïve Bayes performed reasonably well in terms of accuracy, with Logistic Regression achieving a test accuracy of 84/85%, but Naïve Bayes had lower precision (55%) and recall (59%), highlighting its limitations for this application. These results emphasize the importance of selecting an appropriate algorithm based on the specific requirements and constraints of IoT systems, where both accuracy and execution time are critical factors.

2) Dataset 2: [19] This dataset required extensive preprocessing due to its 24 features, including three categorical features with 16 different values. Upon evaluating various machine learning algorithms, it was found that Random Forest had the highest precision (80%) and a training accuracy of 98%, but a moderate test accuracy of 85/85%. Decision Tree showed a high training accuracy of 99% but lower performance on test data with a precision of 51% and a test accuracy of 78/80%. K-NN and SVM had more balanced results, with K-NN [16] achieving a precision of 75% and a test accuracy of 81/84%, while SVM had a precision of 78% and a consistent test accuracy of 85/85%. Notably, K-NN had the longest execution time at 10.90 seconds, followed by SVM at 9.07 seconds, highlighting their computational demands. Logistic Regression and Naïve Bayes showed lower performance, with Logistic Regression having a precision of 47% and a test accuracy of 77/80%, and Naïve Bayes achieving a precision of 55% but with the shortest execution time of 0.05 seconds. These results underscore the trade-offs between accuracy and execution time, emphasizing the importance of preprocessing and algorithm selection in handling complex IoT datasets.

3) Dataset 3: [20] The dataset, consisting of six columns and 112,001 rows, underwent extensive preprocessing to manage its complexity. Evaluating various machine learning algorithms revealed that Random Forest and K-NN achieved the highest precision (85%) and comparable training and test accuracies (95% and 92/92% respectively for Random Forest, 93% and 92/92% for K-NN). Despite their accuracy, Random Forest had the highest execution time at 10.01 seconds, while K-NN [18] executed in 3.24 seconds. SVM also performed well with a precision of 82% and a test accuracy of 92/93%, but had a relatively high execution time of 6.01 seconds. Decision Tree showed a precision of 83% and a test accuracy of 91/91%, with an execution time of 5.1 seconds. Logistic Regression lagged behind significantly with a precision of 25% and a test accuracy of 51/51%, and the highest execution time of 18.2 seconds. Naïve Bayes, while having lower precision (57%), achieved a reasonable test accuracy of 80/82% with the shortest execution time of 0.06 seconds. These results highlight the importance of balancing performance metrics and execution time when selecting algorithms for large IoT datasets.

6.2 Experiment Tables

Table 1. Algorithmic performance analysis on the dataset 1

Algorithms	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Training Set Accuracy (%)**	**Test Set Accuracy (%) (Average/Highest)**	**Execution Time (Second)**
Decision Tree	95	95	95	99	98/99	0.07
K-NN	96	96	96	99	98/99	**0.06**
SVM	96	97	96	99	97/98	2.70
Random Forest	**98**	**98**	**98**	**100**	**98/99**	2.10
Logistic Regression	78	85	81	84	84/85	0.11
Naïve Bayes	55	59	57	90	89/90	1.66

Table 2. Algorithmic performance analysis on the dataset 2

Algorithms	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Training Set Accuracy (%)**	**Test Set Accuracy (%) (Average/Highest)**	**Execution Time (Second)**
Decision Tree	51	53	52	**99**	78/80	1.88
K-NN	75	72	73	90	81/84	10.90
SVM	78	**80**	79	85	85/85	9.07
Random Forest	**80**	**72**	**76**	**98**	**85/85**	1.51
Logistic Regression	47	50	48	77	77/80	6.81
Naïve Bayes	55	59	57	80	80/85	**0.05**

Table 3. Algorithmic performance analysis on the dataset 3

Algorithms	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Training Set Accuracy (%)**	**Test Set Accuracy (%) (Average/Highest)**	**Execution Time (Second)**
Decision Tree	83	78	80	95	91/91	5.1
K-NN	**85**	**82**	**83**	93	92/92	**3.24**
SVM	82	83	82	92	**92/93**	6.01
Random Forest	**85**	79	**82**	**95**	92/92	10.01
Logistic Regression	25	28	26	51	51/51	18.2
Naïve Bayes	57	58	57	80	80/82	**0.06**

7. Conclusion

In this study, we evaluated the performance of six machine learning algorithms (Random Forest, Decision Tree, K-NN, SVM, Logistic Regression, Naive Bayes) for classification tasks in IoT environments using three diverse datasets. Random Forest consistently emerged as the leader in terms of precision, achieving the highest values across all datasets (reaching 98% in Dataset 1). However, its execution time was often the highest, particularly for complex datasets (10.01 seconds in Dataset 3). Decision Tree offered a good accuracy-speed trade-off, performing well in Datasets 1 and 3 but struggling with overfitting in Dataset 2. K-NN demonstrated balanced accuracy and relatively fast execution times, making it suitable for real-time applications where speed is a priority. SVM maintained consistent accuracy but exhibited high execution times, suggesting its use might be more appropriate for non-time-critical tasks. Logistic Regression and Naive Bayes generally showed lower precision, with Naive Bayes excelling in terms of execution speed (always under 0.1 seconds) but falling short in complex scenarios. These findings highlight the importance of considering both accuracy and execution time when selecting machine learning algorithms for IoT applications. The optimal choice depends on the specific requirements of the application, with Random Forest a strong contender for tasks prioritizing high accuracy, while K-NN and Naive Bayes are valuable options when real-time processing and low latency are crucial.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks,* vol. 54, 2010, pp. 2787-2805.

[2] ERC: European Research Cluster of the Internet of Things, Accessed on: Dec. 4, 2019. [Online]. Available: http://www.internet-of-things-research.eu/about_iot.htm.

[3] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile networks and applications,* vol. 19, no. 2, 2014, pp. 171-209.

[4] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature,* vol. 521, no. 7553, 2015, pp. 452-459.

[5] EFD: Elderly Fall Prediction and detection, Accessed on: Mar. 6, 2024 . [online]. Available: https://www.kaggle.com/datasets/laavanya/elderly-fall-prediction-and-detection.

[6] V. Khadse, P. N. Mahalle, and S. V. Biraris, "An Empirical Comparison of Supervised Machine Learning Algorithms for Internet of Things Data," in *Proceedings of the Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1-6.

[7] Z. Xuan, F. Zhubing, L. Liequan, Y. Junwei, and P. Dongmei, "Comparison of four algorithms based on machine learning for cooling load forecasting of large-scale shopping mall," *Energy Procedia,* vol. 142, 2017, pp. 1799-1804.

[8] M. A. Ayu, S. A. Ismail, A. F. A. Matin, and T. Mantoro, "A comparison study of classifier algorithms for mobile-phone's accelerometer-based activity recognition," *Procedia Engineering,* vol. 41, 2012, pp. 224-229.

[9] Y. K. Ever, K. Dimililer, and B. Sekeroglu, "Comparison of Machine Learning Techniques for Prediction Problems," in Workshops of the International Conference on Advanced Information Networking and Applications, 2019, pp. 713-723

[10] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Comparison of machine learning methods for estimating energy consumption in buildings," in *Proceedings of the international conference on probabilistic methods applied to power systems (PMAPS)*, 2014, pp. 1-6.

[11] B. M. H. Alhafidh and W. H. Allen, "Comparison and performance analysis of machine learning algorithms for the prediction of human actions in a smart home environment," in *Proceedings of the International Conference on Compute and Data Analysis*, 2017, pp. 54-59.

[12] F. Almaguer-Angeles, J. Murphy, L. Murphy, and A. O. Portillo-Dominguez, "Choosing Machine Learning Algorithms for Anomaly Detection in Smart Building IoT Scenarios," in *IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 491- 495.

[13] D. Kleyko, R. Hostettler, W. Birk, and E. Osipov, "Comparison of machine learning techniques for vehicle classification using road side sensors," in *Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 572-577.

[14] K. Sabanci, E. Yigit, D. Ustun, A. Toktas, and M. F. Aslan, "WiFi Based Indoor Localization: Application and Comparison of Machine Learning Algorithms," in *23rd International Seminar/Workshop on Direct and*

Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), 2018, pp. 246-251.

[15] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of eight data mining algorithms for smarter Internet of Things (IoT)," *Procedia Computer Science,* vol. 98, 2016, pp. 437-442.

[16] R. Chettri, S. Pradhan, and L. Chettri, "Internet of things: comparative study on classification algorithms (k-nn, naive bayes and case-based reasoning)," *International Journal of Computer Applications,* vol. 130, no. 12, 2015, pp. 7-9.

[17] C. M. Bishop, *Pattern recognition and machine learning*: Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[18] N. S. Altman, "An introduction to kernel and nearest- neighbor nonparametric regression," *The American Statistician,* vol. 46, no. 3, 1992, pp. 175-185.

[19] Rain in Australia dataset, Accessed on: Feb. 5, 2024. [Online]. Available: https://www.kaggle.com/jsphyg/weatherdataset-rattle-package.

[20] EPM: Elevator Predictive Maintenance, Accessed on: Mar. 10, 2024. [online]. Available: https://www.kaggle.com/datasets/shivamb/elevator-predictive-maintenance-dataset