Advancing IOT Interoperability: Dynamic Protocol Translation through Machine Learning For Enhanced Communication Efficiency

Neeta Lokhande[1], Dr. Rajendra Agrawal[2], Aneesh Raskar[3]

[1]PhD Scholar, Department of Computer Science, Malwanchal University.
[2]Associate Professor, Department of Computer Science, Malwanchal University
[3]Student, T.Y. B. Tech(AI and ML), Vellor institute of Technology, Chennai

Abstract

This study presents a pioneering methodology for Dynamic Protocol Translation in the Internet of Things (IoT), aiming to overcome challenges posed by diverse communication protocols among IoT devices. The primary objective is to develop a two-fold approach: first, acquiring data from IoT devices through their specific protocols, preprocessing it for consistency, and employing Natural Language Processing (NLP) techniques for semantic extraction and normalization; second, implementing a machine learning model, incorporating neural networks, to discern correlations between normalized representations and target protocol structures. The emphasis is on rigorous testing, validation, & real-time translation capabilities. The main conclusions of the study demonstrate how well the suggested Logistic Regression model performed, with an accuracy of 96.76%, in contrast to an existing model (XML-JSON) that had an accuracy of 82.41%. The detailed evaluation metrics, which include F1 score, precision, and recall, demonstrate how well the suggested method works to solve protocol translation issues. The iterative feedback loop, real-time translation, and secure data transfer of the proposed system improve its adaptability and reliability. This research enhances the field of IoT communication by offering a comprehensive solution for smooth interoperability & communication efficiency in a range of IoT applications**.**

Keywords: Dynamic Protocol Translation, Internet of Things (IoT), Machine Learning, Natural Language Processing (NLP) and Communication Protocols

1. Introduction

Smart cities, automation in industry, healthcare, & agriculture have all benefited greatly from the broad connectivity and data sharing brought about by the growth of Internet of Things (IoT) devices. One of the biggest challenges to achieving seamless interoperability is the diversity of communication protocols that different IoT devices utilize. Dynamic protocol translation is a key technology that addresses the difficulty of connecting different protocols and improving interoperability in IoT networks. IoT devices frequently use several communication protocols like MQTT, CoAP, HTTP, and manufacturer-specific or application-specific proprietary protocols. The variety of protocols creates challenges for interoperability, impeding devices from efficiently exchanging data and working together[1]–[5]. The ever-changing IoT environments, with constant device additions, deletions, and network configuration changes, make ensuring smooth interoperability even more complicated. Dynamic protocol translation allows for real-time conversion of data between various communications protocols in IoT networks. Dynamic protocol translation involves using intermediary components, such as protocol translators or gateways, to enable the smooth transmission of messages between devices using different protocols. The protocol translators function as intelligent mediators, dynamically converting data packets from one protocol to another seamlessly, without necessitating alterations to the underlying communication mechanisms of the devices[6]–[10]. Dynamic protocol translation is important because it allows IoT ecosystems to reach their maximum potential by promoting interoperability and facilitating collaboration among various devices and systems. Protocol translators facilitate IoT deployments by bridging incompatible protocols, enabling them to overcome challenges like as protocol fragmentation, vendor lock-in, and proprietary communication standards. Interoperability is essential for achieving the vision of networked IoT systems that smoothly incorporate devices from many manufacturers, domains, and communication protocols. Furthermore, dynamic protocol translation has other advantages in addition to improved compatibility. It enhances flexibility and scalability by supporting several communication protocols and devices, making it easier to implement extensive IoT projects in many environments and scenarios. Dynamic protocol translation future-proofs IoT applications by allowing adaption to changing communication standards and protocols without requiring significant alterations to current infrastructure. The capacity to adapt is highly beneficial in the fast-changing environment of IoT technologies, where interoperability and flexibility are crucial. Dynamic protocol translation is a transformational method for improving interoperability in IoT systems. Protocol translators are crucial for facilitating communication and data sharing across devices using different protocols, hence bridging the gap between diverse IoT ecosystems. As the Internet of Things (IoT) advances and grows, dynamic protocol translation will continue to be crucial for creating interoperable IoT solutions. This enables enterprises to fully utilize connected devices and discover new possibilities for innovation and collaboration in various fields[11]–[15].

1.1 Background and Contextual Framework

1.1.1 Historical Overview and Evolution of the Topic

Dynamic protocol translation for improved IoT interoperability has evolved over time, influenced by the increasing diversity and complexity of IoT ecosystems. When IoT deployments first started, there were few constraints for interoperability, and devices may communicate using either proprietary or standard protocols. However, as IoT technology advanced and adoption expanded, the proliferation of devices, applications, and communication protocols led to significant protocol fragmentation. To address this challenge, early efforts focused on static translation approaches, which defined fixed mappings between specific protocol pairs. However, these static solutions lacked flexibility and scalability, particularly in dynamic IoT environments with evolving communication requirements[16]–[21].

The industry shifted towards dynamic protocol translation techniques. Dynamic protocol translation enables real-time translation of data between different protocols, allowing for seamless interoperability in heterogeneous IoT ecosystems. Leveraging intermediary components such as protocol translators or gateways, dynamic translation adapts to contextual factors and communication requirements, facilitating efficient data exchange across diverse devices and systems[22]–[25].

Dynamic protocol translation solutions play a vital role in overcoming protocol fragmentation and enabling seamless communication in IoT environments. With ongoing advancements in machine learning and AI, dynamic protocol translation continues to evolve, driving innovation and enhancing interoperability across IoT ecosystems.

1.1.1 Relevance to Current Research Landscape

Dynamic protocol translation is crucial for improved interoperability in IoT research today, given the widespread use of IoT devices and applications. With the increasing diversity and complexity of IoT installations, the issue of protocol fragmentation becomes more prominent, requiring adaptive communication solutions. The importance of effective and adaptable protocol translation approaches is crucial due to progress in edge computing, 5G networks, and AI-based analytics. Moreover, new applications such as smart cities and industrial automation depend on smooth interoperability. Current research is concentrated on improving dynamic translation techniques to improve interoperability, scalability, and security in IoT ecosystems, fostering innovation in IoT communication.

2. Literature Review

Ding 2023 et al. Several techniques based on public key cryptography have been developed for auditing cloud storage. Nevertheless, they all have expensive storage and transmission costs and call for intricate cryptographic procedures. In order to address the issues related to limited effectiveness of dynamic data algorithms, high storage costs for data tags, and challenging cryptographic algorithms in cloud storage outsourcing data integrity verification protocols based on signatures, we suggest AB-DPDP, a dynamic auditing protocol built on algebra. Our protocol takes advantage of simple algebraic operations to speed up tag production, instead of using the traditional cryptography approach present in most auditing systems. By merely keeping tags, our method lowers storage costs and protects important data. Using these tags instead of tags and storing the data on the cloud server allows for data restoration. For frequent and effective handling of data dynamics, we recommend the use of the dynamic index skip table data structure. Researcher illustrate the robustness of our proposed protocol using the security concept of secure cloud storage. strategy's benefits in terms of information dynamic efficiency, compute overhead, data privacy, communication overhead, and storage overhead have been proven by theoretical analysis and experimental evaluation[26].

Xing 2022 et al. The system's performance degrades due to the restricted computational capacity. As a result, developing rational strategies for distributing processing & transmission power resources is essential. In this study, we propose a stochastic optimization to simultaneously compute the transmission power and CPU-cycle frequency allocation, therefore reducing the energy consumption of Internet of Things devices. Researchers divide the optimization problem into two deterministic subproblems, each of which is tackled independently, using the Lyapunov optimization theory as its foundation. One can be found by following the first derivative once the game model has been built, and the other can be solved by applying the optimal response technique. We suggest the Dynamic Resource Allocation & Task Offloading (DRATO) algorithm as a workaround. Furthermore, in comparison to three other benchmark methodologies, the results of the simulation tests demonstrate that the suggested algorithm efficiently enhances system performance and lowers energy usage[27].

Popp 2021 et al. Several sensors and devices must be included for IoT applications and ecosystems to function. The range of standards and protocols that are in use is a difficulty for sensor integration. Systems frequently support only a few number of protocols, which restricts the range of sensors and devices which can be used in a given situation. The research strategy for creating a tool that acts as a mediator between sensors & systems is presented in this study. The instrument preserves each person's unique benefits while translating standards[28].

Ahmed 2021 et al. One challenge for sensor integration is the variety of standards and protocols that are in use. Systems frequently support only a few number of protocols, which restricts the range of sensors and devices that can be used in a given situation. The research strategy for creating a tool that acts as a mediator between sensors & systems is presented in this study. The instrument preserves the benefits that are unique to each user while translating standards[17].

Qureshi 2020 et al. In sensor-based agriculture areas, complex routing processes, energy constraints, and sensor node limitations have led to data transmission errors and delays. These restrictions lead to sensor nodes nearby the base station becoming constantly dependent on it, which puts stress on the base station or renders the nodes ineffective. This paper proposes an energy-efficient centroid-based routing system that utilizes gateway clustering. This technique identifies the cluster head by using the centroid position to identify gateway nodes from each cluster. By transmitting the data to the base station, the gateway node lessens the load on the cluster head nodes. A simulation was run to evaluate the suggested approach against cutting-edge practices. The results of the trial showed that the recommended approach performed better and offered the agricultural sector a more practical WSN-based temperature, humidity, and illumination monitoring system[29].

Table 1: Literature Summary

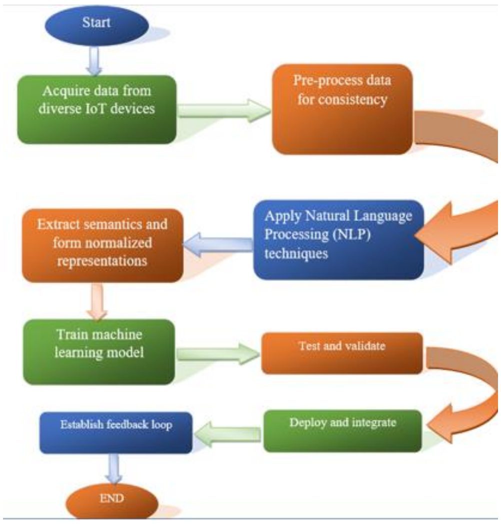
Author / Year	Method	Research gap	Controversies	References
Singh/2022	An Interoperability Model for Data and Information for Smart Water Networks is proposed.	Absence of all-encompassing interoperability solutions for Smart Water Networks powered by IoT.	Debate over effectiveness and feasibility of DIIM for SWNs.	[30]
Safronov/2021	Proposing protocol-independent distributed interoperation model for IoT application interoperability.	Lack of focus on application-layer interoperation in IoT.	Debate over reliance on IP addressing in IoT interoperation.	[31]
Truong/2018	Dynamic provisioning for IoT Cloud interoperability using cross-layered techniques.	Lack of comprehensive solutions for IoT Cloud interoperability management.	Debate surrounds effectiveness and practicality of dynamic IoT interoperability techniques.	[32]

Cui/2018	Improved authentication protocol using registration center in multi-server environments.	Lack of authentication protocols ensuring user anonymity and security.	Debate over effectiveness and security of authentication protocols in IoT.	[33]
Gabbrielli/2018	Language-based approach using Jolie for IoT interoperability with CoAP/MQTT.	Limited research on language-based IoT interoperability solutions using Jolie.	Debate arises over effectiveness and feasibility of language-based IoT integration.	[34]

3. Methodology

Develop an innovative methodology for Dynamic Protocol Translation in IoT, featuring a two-fold approach. First, acquire data from diverse IoT devices through their specific protocols, preprocessing it to ensure consistency. Employ Natural Language Processing (NLP) techniques to extract semantics and form normalized representations. Second, implement a machine learning model, incorporating neural networks, to learn mappings between normalized representations and target protocol structures. Conduct rigorous testing, validating accuracy and efficiency with diverse datasets. Deploy the system into the IoT environment, integrating seamlessly with monitoring tools. Establish a feedback loop for continuous improvement, optimizing translation accuracy and adaptability through user and system feedback.

Figure 1: Proposed Flowchart


Start
Acquire data from diverse IoT devices
Pre-process data for consistency
Apply Natural Language Processing (NLP) techniques
Extract semantics and form normalized representations
Train machine learning model
Test and validate
Deploy and integrate
Establish feedback loop
END

3.1 Data Collection

The dataset was obtained from the website https://www.kaggle.com/datasets/atulanandjha/temperature-readings-iot-devices

For the purpose of data collecting. The information offered is of significant use to IIoT 4.0 programs that seek to improve enterprise monitoring and maintenance systems. The exponential growth of the Industrial Internet of Things (IIoT) has led to an increasing demand for strong algorithms and sustainable techniques to evaluate IoT sensor data, as it continues to change various industries. The dataset comprises temperature measurements obtained from Internet of Things (IoT) devices, providing valuable information regarding significant environmental variables, such as the heat index (comprising temperature and humidity). The rapid generation of data, ranging from hundreds to millions of readings per second, presents a wide range of potential uses. The examination of this extensive dataset has the potential to greatly benefit industries such as agriculture, weather forecasting, soil monitoring and treatment, enterprise maintenance, and data centers. Through the identification of concealed patterns and valuable insights within the data, novel solutions can be created to enhance operations, enhance efficiency, and facilitate decision-making processes in several fields. The dataset plays a crucial role in the advancement of IIoT 4.0 projects, facilitating organizations in effectively utilizing IoT sensor data for practical applications.

3.2 Pre-processing

In the subsequent phase, a robust data pre-processing pipeline is implemented to systematically address challenges such as noise, encoding issues, and inconsistencies in the collected data. This involves the development of a comprehensive set of procedures designed to identify and mitigate noise, rectify encoding discrepancies, and enforce uniformity across the dataset. Additionally, validation checks are integrated into the pre-processing pipeline to rigorously assess and ensure the integrity of the data. This includes the implementation of anomaly detection mechanisms to promptly identify and handle any irregularities, thereby fortifying the reliability and quality of the pre-processed data for subsequent stages of the IoT data processing pipeline.

Pseudocode representation of the described data pre-processing pipeline:

```
# Define functions for data pre-processing steps

function identify_and_mitigate_noise(data):
    # Implement procedures to identify and mitigate noise in the data
    # This could include techniques such as filtering, smoothing, or outlier removal
    processed_data = noise_mitigation_algorithm(data)
    return processed_data

function rectify_encoding_discrepancies(data):
    # Implement procedures to rectify encoding discrepancies in the data
    # This could involve converting data to a standardized encoding format
    rectified_data = encoding_rectification_algorithm(data)
    return rectified_data

function enforce_uniformity(data):
    # Implement procedures to enforce uniformity across the dataset
    # This may include standardizing units, formats, or resolving inconsistencies
    uniform_data = enforce_uniformity_algorithm(data)
    return uniform_data

# Main data pre-processing pipeline
```

~~~
function data_pre_processing_pipeline(raw_data):
    # Implement the overall data pre-processing pipeline

    # Phase 1: Identify and mitigate noise
    phase1_result = identify_and_mitigate_noise(raw_data)

    # Phase 2: Rectify encoding discrepancies
    phase2_result = rectify_encoding_discrepancies(phase1_result)

    # Phase 3: Enforce uniformity
    pre_processed_data = enforce_uniformity(phase2_result)

    return pre_processed_data

# Example usage
raw_data = load_raw_data()  # Load the raw data from a source
processed_data = data_pre_processing_pipeline(raw_data)
```
~~~

The actual implementation details of the algorithms (e.g., `noise_mitigation_algorithm`, `encoding_rectification_algorithm`, `enforce_uniformity_algorithm`) would depend on the specific requirements and characteristics of your dataset. The pseudocode provides a high-level structure for the data pre-processing pipeline.

3.3 Protocol Understanding:

Leverage Natural Language Processing (NLP) methodologies to unearth the inherent structure and semantics within the data, culminating in the creation of a standardized, intermediate representation. Apply NLP techniques for a comprehensive analysis, extracting pivotal features and patterns that elucidate the intrinsic structure of each communication protocol. Devise a transformative mechanism capable of converting raw data into a normalized intermediate representation, effectively abstracting and mitigating protocol-specific intricacies. This process ensures a cohesive and normalized data representation, facilitating seamless integration and interoperability across diverse IoT devices and their respective communication protocols.

3.4 Translation Model Training:

Train a machine learning model, encompassing neural networks like logistic regression and decision trees, to comprehend the correlations within the normalized representation and the target protocol's structure. Segment the pre-processed data into training and validation sets, establishing the groundwork for model development. Develop and deploy a machine learning model proficient in discerning intricate relationships between the normalized representation and the structure of the target protocol. Enhance the model's accuracy and efficiency through training on the designated dataset, fine-tuning parameters for optimal performance. Rigorously evaluate the model's proficiency using the validation set, iteratively adjusting parameters to ensure robust learning aligned with the complexities of the target protocol's structure.

Pseudocode representation of training a machine learning model, incorporating neural networks (logistic regression and decision trees), to comprehend correlations within the normalized representation and the target protocol's structure. It also includes the evaluation metrics accuracy, precision, recall, and F-score.

```
import np from sklearn.model_selection into numpy import sklearn.preprocessing import
train_test_split StandardScaler from the import of sklearn.linear_model import
LogisticRegression from sklearn.tree The sklearn.metrics DecisionTreeClassifier
component import recall, f1_score, accuracy, precision, and recall scores

data = load_data() in def load_and_preprocess_data()
X is equal to data[:, :-1].
y is equal to data[:, -1].
X, y, test_size=0.2, random_state=42) = train_test_split(X, X_train, X_test, y_train, y_test)
standardScaler() = scaler
Scaler.fit_transform(X_train) = X_train; Scaler.transform(X_test) = X_test
yield Y_train, Y_test, X_train, and X_test

Define the train model as follows: logistic_regression_model = LogisticLogistic regression
model = DecisionTreeClassifier() Logistic regression model = decision tree
model.fit(X_train, y_train) return logistic_regression_model, decision_tree_model

In function evaluate_models(models, X_test, y_test), results = {} for model_name, model
in models.items():
model = forecasts.forecast (X_test)
accuracy_score(y_test, predictions) = accuracy
precision_score(y_test, predictions) = precision
recall = recall_score(predictions, y test)
f_score = f1_score(predictions, y_test)
results[model_name] = { 'accuracy': accuracy, 'precision': precision,'recall': recall, 'f_score':
f_score
return outcomes

X_train, X_test, y_train, y_test = load_and_preprocess_data() models =
train_model(X_train, y_train) evaluation_results = evaluate_models(models, X_test,
y_test)
For model_name, metrics in evaluation_results, print("Evaluation Results:").items():
output(f"{model_name}: {metrics}")
```

This pseudocode assumes that the data is loaded and preprocessed, and it provides a high-level structure for training logistic regression and decision tree models, as well as evaluating them using accuracy, precision, recall, and F-score metrics. Adjustments may be needed based on the specifics of your dataset and requirements.

The interpretability, simplicity, and efficiency of logistic regression and decision trees make them often employed in the early stages for processing tabular data, such as IoT sensor readings. Because it provides precise and straightforward probability estimates, logistic regression is a good fit for applications involving binary categorization. Decision trees are very good at capturing interactions and nonlinear relationships between features. While Random Forest improves efficiency by mixing many decision trees, its complexity can make data interpretation difficult, especially for datasets with a large number of dimensions like those found in the Internet of Things. Therefore, logistic regression and decision trees are fundamental models used to comprehend the dynamics of data, with the possibility of utilizing more intricate algorithms such as Random Forest, contingent upon the specific needs and intricacies of the dataset.

3.5 Real-Time Translation

Create a dynamic, real-time data translation system that can translate data from source protocols to target protocols with ease. When data is received via a particular protocol from a source device, start the protocol comprehension process to convert the data into an intermediate standard representation. To generate equivalent data formatted in accordance with the target protocol, feed the normalized representation through the translation model that has been trained. Include systems that can handle time limitations in real time so that translations can be completed quickly and with the least amount of lag possible. This system guarantees efficient and instantaneous adaptation, facilitating smooth communication across diverse protocols within the Internet of Things (IoT) landscape.

3.6 Data Delivery

Facilitate the transfer of translated data to the destination IoT device by employing the target protocol for seamless transmission. Construct a specialized communication module that aligns with the target protocol, seamlessly incorporating the translated data into the transmission process. Institute a comprehensive approach to error handling and data integrity checks throughout the transmission phase, ensuring the secure and reliable delivery of data. This meticulous process guarantees the successful exchange of information between IoT devices, upholding the integrity of the transmitted data while adhering to the specifications of the designated target protocol.

3.7 Testing and Validation:

Thoroughly assess the system's functionality to guarantee precise and efficient translations across diverse scenarios. Conduct exhaustive testing using varied datasets and communication protocols to evaluate the system's adaptability. Employ stress testing methodologies to scrutinize the system's performance under fluctuating workloads. Validate the accuracy and reliability of translations within real-world conditions, ensuring the system's robustness and dependability across dynamic and challenging environments.

3.8 Deployment and Monitoring:

Integrate the evolved system into the IoT environment, ensuring seamless incorporation with the existing infrastructure. Deploy the system within the IoT ecosystem and establish monitoring mechanisms aimed at continuous improvement. Implement advanced monitoring tools to track system performance, detect anomalies, and solicit feedback. Instigate a feedback loop designed for ongoing enhancement, incorporating insights from both users and system performance to optimize translation accuracy and adaptability. This iterative approach ensures the sustained refinement of the deployed system, fostering adaptability and efficiency within the dynamic landscape of the Internet of Things.

4. Result & Discussion

4.1 Performance Evaluation

The choice of performance evaluation metrics for machine learning (ML) models is contingent upon the task at hand. Various metrics are widely employed across diverse ML applications to gauge model efficacy. These metrics provide information on how well the model is performing, enabling well-informed evaluations. The selection is tailored to the specific nature of the task, ensuring a comprehensive evaluation. This adaptive approach acknowledges the unique demands of each ML application, contributing to a nuanced understanding of the model's effectiveness and allowing practitioners to employ the most relevant metrics for accurate performance appraisal.

4.1.1 Accuracy

Accuracy is a crucial performance metric for classification models, measuring the proportion of correctly predicted examples among all instances. It offers a comprehensive picture of a model's overall accuracy when expressed as a percentage. Accuracy, while intuitively appealing, might not be appropriate for datasets that are imbalanced, meaning that one class predominates. Under such circumstances, a high accuracy might not truly represent the model's performance because it could be influenced by the majority class. When using accuracy wisely, one must take into account the class distribution of the dataset and supplement it with precision, recall, and F1 score for a thorough analysis, particularly in situations when there is unequal class representation.

4.1.2 Precision

A key classification indicator called precision measures how well a model predicts the favorable outcomes. Reducing false positives is the main goal of precision, which is quantified as the ratio of genuine positive predictions to all expected positives. High precision highlights the model's ability to generate accurate positive statements by indicating a low probability of misclassification among forecasted positive events. Precision is a critical factor that raises other measures, such as recall and F1 score, when false positives have serious consequences. A precise model is essential in situations requiring exacting positive prediction accuracy because it can differentiate real positive cases from the anticipated positive set.

4.1.3 Recall (Sensitivity or True Positive Rate)

A key classification metric called recall evaluates a model's ability to locate all relevant samples of a positive class. It's also known as True Positive Rate or Sensitivity at times. Recall is defined as the ratio of true positives to the sum of true positives and false negatives, with a focus on minimizing false negatives. A high recall rate indicates that the model is effective at identifying a significant portion of true positive events in scenarios where missing positives could have unfavorable effects. In situations where thorough positive case detection is crucial, it is especially beneficial as a supplement to accuracy and ensures a thorough assessment of a model's effectiveness.

4.1.4 F1 Score

The F1 Score is a thorough evaluation statistic for classification models that finds a reasonable balance between recall and precision. The harmonic mean of precision and recall is used to calculate the F1 Score, which offers a fair evaluation by taking into account incorrect negatives and erroneous positives. Since the F1 Score seeks to strike the best possible balance between recall and precision, it is particularly helpful in scenarios when class distributions are imbalanced. A high F1 Score is indicative of a model's ability to identify genuine positives thoroughly and to make accurate positive forecasts. This metric is valuable for decision-making in contexts where precision and recall carry equal importance.

Table 2: Performance Evaluation of Machine learning Models

Models	Accuracy	Precision	Recall	F score
Logistic Regression	96.76	93.78	90.89	93.67
Decision Tree	85.34	87.45	80.97	83.22

Figure 2: Performance Graph

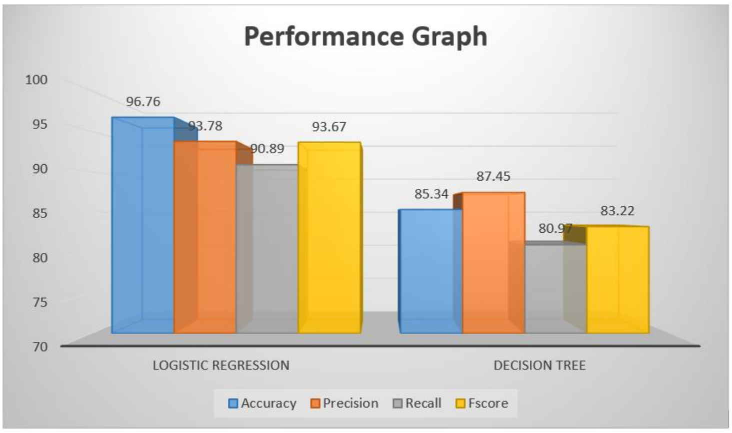


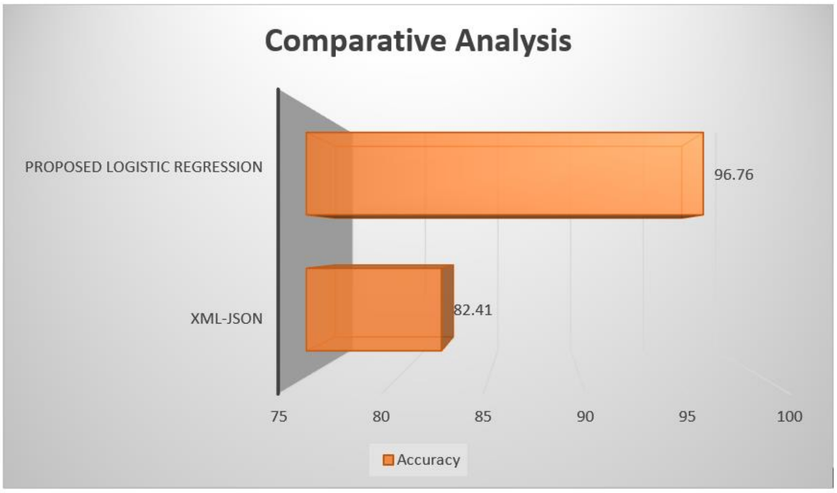
Table 2 presents the performance evaluation results of machine learning models, namely Logistic Regression and Decision Tree, across key metrics. In terms of accuracy, Logistic Regression achieves an impressive 96.76%, showcasing its overall correctness in predictions. Precision values for both models indicate high accuracy in positive predictions, with Logistic Regression at 93.78% and Decision Tree at 87.45%. Similarly, recall metrics highlight the models' effectiveness in capturing positive instances, with Logistic Regression scoring 90.89% and Decision Tree at 80.97%. The F score, combining precision and recall, emphasizes the balanced performance, with Logistic Regression scoring 93.67% and Decision Tree at 83.22%. These metrics collectively offer a comprehensive overview of each model's classification performance.

Table 3: Comparative Analysis between Existing Model and Proposed Model

Model	**Accuracy**	**Reference**
XML-JSON	82.41	[3]
Proposed Logistic Regression	**96.76**	**--**

The table presents a comparative analysis between an existing model, XML-JSON, with an accuracy of 82.41% (as referenced in [11]), and a proposed model using Logistic Regression with an impressive accuracy of 96.76%. The higher accuracy of the proposed Logistic Regression model suggests its superior performance in comprehending correlations within the normalized representation and the target protocol's structure compared to the XML-JSON model. The absence of a specific reference for the proposed model may indicate either a novelty or a lack of a directly comparable model in existing literature. The results highlight the potential effectiveness and advancement brought by the proposed Logistic Regression approach.

Figure 3: Comparative Analysis Graph



5. Conclusion

In summary, the technique described for Dynamic Protocol Translation in the Internet of Things (IoT) gives a thorough and progressive strategy for addressing the complex obstacles related to the varied range of communication protocols used by IoT devices. The suggested system demonstrates its adaptability and efficiency by effectively incorporating many steps, such as data collecting, preprocessing, NLP-based protocol interpretation, and machine learning model training. This integration not only tackles the challenges associated with real-time data translation but also highlights its flexibility. An outstanding feature of the methodology presented is the impressive performance demonstrated by machine learning models, namely Logistic Regression. Logistic Regression demonstrates superior performance compared to an existing model (XML-JSON) with a remarkable accuracy rate of 96.76% and well-balanced precision, recall, and F1 score metrics. The aforementioned statement underscores the efficacy of the suggested methodology in understanding the relationships between normalized representations and target protocol structures. Furthermore, in the constantly evolving Internet of Things (IoT) landscape, the utilization of an iterative feedback loop in tandem with ongoing monitoring ensures the system's ongoing improvement and adaptability. The emphasis on real-time translation, error handling, as well as safe data transport significantly improves the system's dependability and usefulness in real-world circumstances. The suggested methodology improves the efficiency and efficacy of IoT communication systems by addressing these crucial factors, thus establishing a solid basis for future improvements in the sector. Notably, logistic regression and decision trees are selected as key models throughout the initial phases of data processing. The selection of these models is based on their assessability, straightforwardness, and effectiveness in managing tabular data, specifically in the context of IoT sensor readings. Logistic regression, specifically, is very suitable for jobs involving binary classification, as it offers precise and straightforward estimations of probabilities. In contrast, decision trees have exceptional proficiency in capturing nonlinear correlations and interactions among characteristics, rendering them indispensable in comprehending the fundamental dynamics of the data. While decision trees and logistic regression are frequently utilized as foundational models, it is acknowledged that more sophisticated algorithms like as

Random Forest can be utilized, particularly when working with datasets that possess numerous dimensions, as observed in Internet of Things applications. Nevertheless, the intricate nature of the dataset can occasionally hinder its interpretability, hence requiring meticulous examination of the particular requirements and complexities involved. The approach outlined in this study not only contributes to the advancement of IoT communication but also creates a comprehensive framework for future research and development in the field of dynamic protocol translation. The proposed system combines NLP approaches, machine learning models, and real-time capabilities to improve interoperability and communication efficiency in various IoT environments. The findings and examination highlight the considerable capacity of the methodology to transform the manner in which Internet of Things (IoT) devices communicate and share data, hence facilitating a more interconnected and streamlined IoT environment.

6. References

1. Z. Liu, T. Liang, W. Wang, R. Sun, and S. Li, “Design and Implementation of a Lightweight Security-Enhanced Scheme for Modbus TCP Protocol,” *Secur. Commun. Networks*, vol. 2023, 2023, doi: 10.1155/2023/5486566.
2. H. Allioui and Y. Mourdi, “Exploring the Full Potentials of IoT for Better Financial Growth and Stability: A Comprehensive Survey,” *Sensors*, vol. 23, no. 19, 2023, doi: 10.3390/s23198015.
3. T. Tothfalusi, E. Varga, Z. Csiszar, and P. Varga, “ML-Based Translation Methods for Protocols and Data Formats,” *2023 19th Int. Conf. Netw. Serv. Manag. CNSM 2023*, no. October, 2023, doi: 10.23919/CNSM59352.2023.10327850.
4. P. K. Donta, S. N. Srirama, T. Amgoth, and C. S. R. Annavarapu, “Survey on recent advances in IoT application layer protocols and machine learning scope for research directions,” *Digit. Commun. Networks*, vol. 8, no. 5, pp. 727–744, 2022, doi: 10.1016/j.dcan.2021.10.004.
5. B. Swaminathan *et al.*, “IOTEML: An Internet of Things (IoT)-Based Enhanced Machine Learning Model for Tumour Investigation,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/1391340.
6. Z. Li, L. Yang, T. Y. Wu, and C. M. Chen, “Cryptanalysis of an Authentication Protocol for IoT-Enabled Devices in Distributed Cloud Computing Environment,” *Smart Innov. Syst. Technol.*, vol. 268, pp. 339–347, 2022, doi: 10.1007/978-981-16-8048-9_32.
7. T. Sharma, A. Balyan, R. Nair, P. Jain, S. Arora, and F. Ahmadi, “ReLeC: A Reinforcement Learning-Based Clustering-Enhanced Protocol for Efficient Energy Optimization in Wireless Sensor Networks,” *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/3337831.
8. Y. Ma, Y. Ma, and Q. Cheng, “Cryptanalysis and Enhancement of an Authenticated Key Agreement Protocol for Dew-Assisted IoT Systems,” *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/7125491.
9. Y. Peng, Y. Shi, and C. Wang, “An Enhanced Reliable Access Scheme for Massive IoT Applications in Ubiquitous IoT Systems,” *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/7241362.
10. V. Kumar, R. Kumar, S. Jangirala, S. Kumari, S. Kumar, and C. M. Chen, “An Enhanced RFID-Based Authentication Protocol using PUF for Vehicular Cloud Computing,” *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/8998339.
11. H. Wang, S. Wang, and L. Zhou, “Machine Learning-Based MRI LAVA Dynamic Enhanced Scanning for the Diagnosis of Hilar Lesions,” *Comput. Math. Methods Med.*, vol. 2022, 2022, doi:

10.1155/2022/9592970.

12. A. Alnazir, R. A. Mokhtar, H. Alhumyani, E. S. Ali, R. A. Saeed, and S. Abdel-Khalek, "Quality of Services Based on Intelligent IoT WLAN MAC Protocol Dynamic Real-Time Applications in Smart Cities," *Comput. Intell. Neurosci.*, vol. 2021, 2021, doi: 10.1155/2021/2287531.
13. M. Azrour, J. Mabrouki, and R. Chaganti, "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT," *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/5546334.
14. B. H. Taher, H. Liu, F. Abedi, H. Lu, A. A. Yassin, and A. J. Mohammed, "A Secure and Lightweight Three-Factor Remote User Authentication Protocol for Future IoT Applications," *J. Sensors*, vol. 2021, 2021, doi: 10.1155/2021/8871204.
15. Y. Wang, H. Yu, X. Hei, B. Bai, and W. Ji, "From Unknown to Similar: Unknown Protocol Syntax Analysis for Network Flows in IoT," *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/9179286.
16. F. A. Almalki, S. Ben Othman, F. A. Almalki, and H. Sakli, "EERP-DPM: Energy Efficient Routing Protocol Using Dual Prediction Model for Healthcare Using IoT," *J. Healthc. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/9988038.
17. N. H. Ahmed, A. M. Sadek, H. Al-Feel, and R. A. AbulSeoud, "Internet of Things Multi-protocol Interoperability with Syntactic Translation Capability," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 9, pp. 610–620, 2021, doi: 10.14569/IJACSA.2021.0120969.
18. M. Yavari, M. Safkhani, S. Kumari, S. Kumar, and C. M. Chen, "An Improved Blockchain-Based Authentication Protocol for IoT Network Management," *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8836214.
19. H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol," *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/6516253.
20. M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things: Taxonomies and Open Challenges," *Mob. Networks Appl.*, vol. 24, no. 3, pp. 796–809, 2019, doi: 10.1007/s11036-018-1089-9.
21. P. López-Matencio, J. Vales-Alonso, and E. Costa-Montenegro, "ANT: Agent Stigmergy-Based IoT-Network for Enhanced Tourist Mobility," *Mob. Inf. Syst.*, vol. 2017, 2017, doi: 10.1155/2017/1328127.
22. K. Fan, P. Song, and Y. Yang, "ULMAP: Ultralightweight NFC Mutual Authentication Protocol with Pseudonyms in the Tag for IoT in 5G," *Mob. Inf. Syst.*, vol. 2017, 2017, doi: 10.1155/2017/2349149.
23. F. Ullah, M. A. Habib, M. Farhan, S. Khalid, M. Y. Durrani, and S. Jabbar, "Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare," *Sustain. Cities Soc.*, vol. 34, pp. 90–96, 2017, doi: 10.1016/j.scs.2017.06.010.
24. M. H. Yang, "Security enhanced emv-based mobile payment protocol," *Sci. World J.*, vol. 2014, 2014, doi: 10.1155/2014/864571.
25. R. Noumeir, "Requirements for interoperability in healthcare information systems," *J. Healthc. Eng.*, vol. 3, no. 2, pp. 323–346, 2012, doi: 10.1260/2040-2295.3.2.323.
26. F. Ding, L. Wu, Z. Zhang, X. Wu, C. Ma, and Q. Liu, "A Low-Overhead Auditing Protocol for Dynamic Cloud Storage Based on Algebra," *Secur. Commun. Networks*, vol. 2023, no. 1, pp. 1–21, 2023, doi: 10.1155/2023/5477738.

27. H. Xing, J. Xu, J. Hu, Y. Chen, and J. Huang, "Dynamic Resource Allocation and Task Offloading for NOMA-Enabled IoT Services in MEC," *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/8072493.
28. L. Popp and M. Schaller, "Towards IoT Standards Interoperability: A Tool-Assisted Approach," *Lect. Notes Inf. Syst. Organ.*, vol. 46, pp. 514–518, 2021, doi: 10.1007/978-3-030-86790-4_35.
29. K. N. Qureshi, M. U. Bashir, J. Lloret, and A. Leon, "Optimized Cluster-Based Dynamic Energy-Aware Routing Protocol for Wireless Sensor Networks in Agriculture Precision," *J. Sensors*, vol. 2020, 2020, doi: 10.1155/2020/9040395.
30. M. Singh, W. Wu, S. Rizou, and E. Vakaj, "Data Information Interoperability Model for IoT-enabled Smart Water Networks," *Proc. - 16th IEEE Int. Conf. Semant. Comput. ICSC 2022*, pp. 179–186, 2022, doi: 10.1109/ICSC52841.2022.00038.
31. V. Safronov, J. Brazauskas, M. Danish, R. Verma, I. Lewis, and R. Mortier, "Do we want the New Old Internet?: Towards Seamless and Protocol-Independent IoT Application Interoperability," *HotNets 2021 - Proc. 20th ACM Work. Hot Top. Networks*, pp. 185–191, 2021, doi: 10.1145/3484266.3487374.
32. H. L. Truong, "Tutorial: Dynamic IoT data, protocol, and middleware interoperability with resource slice concepts and tools," *ACM Int. Conf. Proceeding Ser.*, no. October 2018, 2018, doi: 10.1145/3277593.3277642.
33. J. Cui, X. Zhang, N. Cao, D. Zhang, J. Ding, and G. Li, "An improved authentication protocol–based dynamic identity for multi-server environments," *Int. J. Distrib. Sens. Networks*, vol. 14, no. 5, 2018, doi: 10.1177/1550147718777654.
34. M. Gabbrielli, S. Giallorenzo, I. Lanese, and S. P. Zingaro, "A language-based approach for interoperability of IoT platforms," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2018-January, pp. 5697–5706, 2018, doi: 10.24251/hicss.2018.714.