RECOGNIZED IN "Y" CATEGORY BY

ISSN Print: 3005-8007
ISSN Online: 3005-8015
Volume 3
Issue 1
January - June 2025

UCP Journal of Engineering & Information Technology

University of Central Punjab

ISSN:
3005-8015 (Online)
3005-8007 (Print)
Vol. 3, Issue 1
(January – June 2025)

(UCP-JEIT)
UCP Journal of Engineering & Information Technology
HEC Recognized (Y- Category)

Volume 3
Issue 1

Faculty of Information Technology & Computer Sciences
&
Faculty of Engineering

University of Central Punjab, Lahore, Pakistan

Editorial Board

Advisory Board

International Members

Dr. Haris Javaid
(AMD, Singapore)

Dr. Demostenes Zegarra Rodriguez
(Federal University of Lavras, Brazil)

Dr. Salman Azhar
(Auburn University, USA)

Dr. Ali Kashif Bashir
(Manchester Metropolitan University, UK)

Dr. Muhammad Ramzan
(Saudi Electronic University, KSA)

Dr. Nasir Rajpoot
(University of Warwick, UK)

Dr. Agnes Jocher
(Technical University of Munich)

Dr. Ali Nasir
(King Fahad University of Petroleum and Minerals)

Dr. Moez Ben Houidi
(King Abdullah University of Science and Technology)

National Members

Subscription Charges National: PKR 1000 per issue
International: US$ 200 per issue

Acknowledgment

The Editorial Board of the UCP Journal of Engineering and Information Technology extends heartfelt appreciation to all those who have played crucial roles in bringing Volume 3, Issue 1 to fruition. We sincerely recognize the invaluable contributions of our esteemed researchers/authors, whose dedication to advancing knowledge has enriched this inaugural edition.

We also extend our gratitude to the diligent reviewers whose expertise and insightful feedback have ensured the quality and rigor of the articles published herein. Your commitment to the peer-review process is deeply valued.

Furthermore, we thank all individuals involved in the publication process, including editorial staff, copyeditors, and designers, whose unwavering support and tireless efforts have been indispensable.

Without the collective dedication of these individuals, the publication of Volume 3, Issue 1 of the UCP Journal of Engineering and Information Technology would not have been possible. We anticipate continued collaboration and the exploration of new frontiers in the realm of engineering and information technology.

Warm regards,
Dr. Muhammad Amjad Iqbal
Editor-in-Chief
UCP Journal of Engineering and Information Technology

Disclaimer

The views expressed in these articles are solely those of the respective authors and do not necessarily reflect the views of the Editorial Board or the management and staff of the University of Central Punjab. While every effort has been made to ensure the accuracy of the information provided by the authors, the Editorial Board does not accept any responsibility for any errors or omissions or breach of copyrights, if any.

Every effort has been made to ensure the accuracy and reliability of the information presented in the articles. However, the Editorial Board and the University of Central Punjab make no representations or warranties regarding the completeness, accuracy, or suitability of the content. Readers are encouraged to exercise their judgment and discretion when interpreting and applying the information contained in these articles.

The UCP Journal of Engineering & Technology is committed to upholding the highest standards of academic integrity and ethical publishing practices. Any concerns, questions, or requests for clarification related to the content published in this journal should be directed to the respective authors, who bear full responsibility for their work.

We appreciate your understanding of this disclaimer and hope that you find the content within this journal informative and thought-provoking.

UCP JOURNAL OF ENGINEERING AND TECHNOLOGY

Table of Contents

Date available online: 23-07-2025
Vol. 3, Issue 1 (January – June 2025)

DOI: https://doi.org/10.24312/ucp-jeit.03.01.444

Securing Financial Transactions: Leveraging Random Forest for Credit Card Fraud Detection

Abu Bakr Qazi[1], Arfan Ali Nagra[1], Khola Farooq[1], Muhammad Yousif[2], Muhammad Haseeb Zia [3]
[1] Department of Computer Science, Lahore Garrison University Lahore Pakistan
[2] School of Computer Science, Minhaj University Lahore Pakistan
[3] Department of Software Engineering Lahore Garrison University Lahore Pakistan

Corresponding author: Muhammad Haseeb Zia (e-mail: haseenzia@lgu.edu.pk).

ABSTRACT
It is a crucial responsibility in the financial sector to safeguard clients and businesses against financial losses by verification of credit cards. It is demonstrated that machine learning algorithms like random forest methods can detect fraud. Multiple decision trees are used in the Random Forest to create predictions. It is very suitable for fraud detection in credit card fraud due to its prowess with high-dimensional and huge datasets. With the help of training on labelled data and performance analysis, the Random Forest algorithm can accurately detect fraudulent transactions and help decrease financial risks. Due to its features of interpretability and durability against overfitting, it is well well-suited tool for firms wanting to improve their fraud detection systems due to. With the use of the strengths of the Random Forest algorithm, accurate analysis and categorization of complex patterns and minute abnormalities present in credit card transactions has become feasible. This strategy helps in the efficient and on-time identification of potential dangers which enables proactive measures to guard against fraudulent activity and increases the accuracy of detection. With this advantage of Random forests, institutions of finance can get an in-depth analysis of the causes that impact fraud incidents and enable them to enhance and improve their security systems. Despite of shifting dynamics of transactions and fraud strategies, the model can tackle overfitting and guarantees reliable performance. Therefore, with the addition of the Random Forest algorithm into card fraud detection frameworks, enterprises acquire the ability to strengthen their defense system and enhance the security and trust of both their clients and the overall financial ecosystem.

INDEX TERMS Random Forest Algorithm, Ensemble Learning, Credit Card, Defence Fortification, Fraud

I. INTRODUCTION
With the advancement of technology fraud in credit cards has grown to be a big issue in the financial sector that affects cardholders as well as financial institutions. To reduce financial losses and maintain the credibility of the system it is important to identify fraudulent transactions as early as possible [1]. With time fraud schemes are advancing in terms of complexity and sophistication which is why conventional rule-based approaches and manual monitoring have reached their limits [2]. RF algorithm is a learning method that uses the prediction of various decision trees and happens to be a very efficient algorithm in this aspect. The model has its versatility and handles big and complicated datasets efficiently and due to its robustness against over-fitting, Random Forest has been widely used in many fields. The model has generated promising results in detecting fraud in credit cards [3].

Over time with the advantage of the Random Forest technique decision trees are built and every tree is trained using a different set of sample data and feature set. Occurrence of fraudulent transactions and various characteristics form patterns. The system is trained by discovering these patterns and connections.

There are many advantages of using a random forest algorithm in detecting frauds of credit cards. First of all, it can manage unbalanced datasets, that have a high proportion of fraudulent transactions than genuine ones, by offering accurate predictions for both groups. In addition to that RF (Random Forest) provides a feature importance evaluation that enables investigators to identify the most important factors influencing fraud detection. The model is descriptive and enables analysts to understand the decision-making process efficiently [5].

This work explores the use of the random forest in detecting credit card fraud. The transaction data of credit cards is pre-processed, features selection is done and then the Random Forest model is trained on labeled data. After the training, the model is assessed with the use of pertinent performance measures. It is anticipated that reliable and effective identification of fraudulent transactions is done by utilizing the strengths of the Random Forest algorithm and it boosts the security and overall integrity of credit card systems [6].

In general, using the Random Forest algorithm to detect credit card fraud shows potential for enhancing fraud detection precision and lowering false positives, improving security for financial institutions and consumers against fraudulent actions.

II. LITERTURE REVIEW
The article talks about how the growing usage of the internet has affected online card transactions and how this has become a reason for an increase in fraud in the

worldwide banking industry. Due to their static nature, traditional rule-based systems have shown to be ineffective at identifying fresh and unreported threats. In response, academics have concentrated on creating systems that use machine learning, particularly deep learning, to identify fraud in an adaptable manner. The development of robust models was hampered by the incomplete understanding of fraudulent card transaction features in earlier investigations. The authors created a dataset of 4 billion non-fraudulent and 245,000 fraudulent transactions from 35 banks in Turkey to fill this gap. They presented and assessed various models for fraud detection based on profiles like models based on type of cards, transaction characteristics, and amount. The efficiency of models against ageing and zero-day attacks was demonstrated through temporal and spatial analysis. By utilizing sophisticated profiling approaches and examining transaction patterns, this paper ultimately aims to improve fraud detection in online card transactions, aiding in the creation of more reliable and resilient fraud detection models. [1]

Credit card usage has significantly increased in the digital economy, which has increased credit card fraud, as this study explores. Although ML algorithms are already being used to identify credit card fraud, these algorithms struggle because of the constantly changing shopping habits of customers and the dataset's class imbalance issue.[2]

The effectiveness of transaction fraud detection techniques is examined, as well as how they affect user losses in online transactions. There is a high rate of misjudgment since it is difficult for current approaches to adequately de-scribe the transaction behaviours of low-frequency users with small transaction volumes. To improve accuracy for low-frequency users, the research presents a novel approach that generates individual transaction behaviours. To in-crease accuracy, this approach transfers the current transaction group behaviour and transaction status. With the help of previous transaction history benchmarks for users' unique transaction behavior is constructed by identifying the ideal risk threshold. To construct the common behavior of the current transaction group, behavioral traits from authentic as well as fraudulent samples are extracted by using the DBSCAN clustering algorithm. With the use of the sliding window method, the current transaction status is extracted from transaction records. A new transaction be-haviour is provided to the user by integrating these elements and with the use of the Naive Bayes method a multi-behavior detection model is recommended to determine the chances of a transaction being fraudulent. This method is suitable for low-frequency users. As the experimental results show the method effectively identifies fraud-lent transactions with a low misjudgment rate for genuine transactions.[3]

This research work explores the challenges that are involved in ensuring the trustworthiness and dependability of transactions in the autonomous as well as open environment of e-commerce in online social networks (ECOS). There is no guarantee of privacy or protection against fraud in ECOS network transactions. As an efficient solution to this problem, it is suggested that trust management strategies should be used.

To tackle the problem of trust in ECOS, this research work suggests a hybrid trust paradigm that is based on factor enrichment. The architecture suggested in this work uses three levels of trust to create a reliable view among people involved in the transaction. It defines private reputation as a dynamically changing perception of reliability. Moreover, it adds that common reputation is a group-wide, transferable factor of trust. It is strengthened by characteristics that increase reliability and consistency. In addition to that hybrid trust combines personal and public reputation to pro-vide cohesive and reliable impressions. Privacy and anti-fraud criteria are catered by the hybrid trust model to further evaluate the security of transactions. [4]

The results of this study show the performance and depict how well it can handle problems related to credit card data like imbalanced distribution of classes and overlapping class samples. [5]

This article states that credit card fraud is now a major problem in e-commerce technologies and caused businesses to suffer large financial losses. Ultimately these frauds prompt the creation of efficient fraud detection systems. It is a difficult task to accurately detect fraud because of the limitations of conventional machine learning methods and in-sufficient credit card information.

The proposed technique is compared with several algorithms like support vector machine (SVM), multi-layer perceptron (MLP), conventional AdaBoost, decision tree and LSTM to verify its performance. According to experimental findings, classifiers trained using resampled data perform better than those trained without it. The suggested LSTM ensemble classifier outperforms the previous techniques, with a sensitivity of 0.996 and a specificity of 0.998. [6]

The essay highlights the growing need for cutting-edge fraud prevention techniques in a cashless world, especially in light of the sizeable predicted global loss brought on by fraud.

By dividing users into old and new users, the research offers the idea of user separation rather than just concentrating on foretelling illegal transactions. For each user group, distinct models—CatBoost and Deep Neural Net-works—are used. The work also presents numerous ways to improve detection accuracy including dealing with the issue of imbalanced datasets, feature modification and feature engineering. The Deep Neural Network model achieved an AUC of 0.84 whereas the CatBoost model obtained a score of 0.97. That is why using these scores, one would expect counterfeit credit cards with higher accuracy. It gives an understanding of the user separation method for the detection of credit card frauds that employs CatBoost and a deep learning approach. Results demonstrate how effectively the

algorithm detects fake credit cards and highlight the methods that can be used to improve the precision of the detection. [7]

To enhance the efficiency, security as well as the effectiveness of the quickly growing private insurance business, this work calls for the adoption of technology. Traditional methods which rely on people are slow and can contain errors since they are not very efficient. To conclude, this paper provides a framework for a safe and automated insurance system to tackle these problems This approach tries to reduce losses, reduce contact with people, enhance secu-rity, indicate clients with high risks, and detect fake claims. It utilizes blockchain technology for secure transactions and data exchange among the insurance network agents. Also, it employs the extreme gradient boosting (XGBoost) method which is more effective than the rest in terms of performance and especially in detecting fake claims. In ag-gregate, by integrating blockchain technology and machine learning an extensive analytical framework can be turned into a set of powerful tools with benefits in terms of increased efficiency, enhanced accuracy, and enhanced protection of the insurance market.[8]

The paper is based on the issue of credit card fraud in electronic commerce systems and introduces. OLightGBM (Optimised Light Gradient Boosting Machine) is one of the latest smart techniques to detect fraud. With proper tuning of hyperparameters of LightGBM, the Bayesian-based hyperparameter optimisation also helps in fraud detection. From the testing carried out with real data, it's important to know that OLightGBM outperforms the other models. The proposed method acquires a precision of 97.34%, an AUC of 92.88% and an F1-score of 56.95%. It also achieved an accuracy of 98.40%. In conclusion, the study recommends a viable approach toward card fraudulent detection of credit cards.Transactions and reduce financial losses that use LightGBM and Bayesian-based hyperparameter-tuned optimization. [9]

The paper aims to cope with the issues related to the identification of fraud in online shopping as well as the in-creasing concern with it. Because of this, sophisticated deep-learning algorithms are necessary for utilization. The problem with applying machine learning techniques is that the calculations have finite accuracy. The article conducts a comparative analysis with the use of dataset i:e European Card Benchmark and shows appreciable enhancements in convolutional neural network architectures' fraud detection precision, accuracy, f1-score and AUC curves. The suggested models outperform conventional machine learning techniques and exhibit their efficacy in actual situations involving the detection of frauds of credit cards. The paper offers a thorough method that makes use of deep learning algorithms for more accurate and trustworthy credit card fraud detection. Overall, the research work demonstrates the suggested models' higher performance in comparison to conventional ML techniques [10].

To automate the detection process and decrease the number of false positive alerts, the article analyses the difficulties associated with the frequency of false positives in detection systems and advises the use of deep neural networks. Although sophisticated techniques like statistical modelling and models of machine learning are used by fraud detection systems, human intervention is still necessary to confirm fraud instances or rule out false positives, which results in inefficiencies and increased costs.

The study investigates how to interpret alerts produced by a fraud detection system using deep learning techniques, particularly deep neural networks, and assesses how accurate these techniques are at detecting false positives. We evaluate and contrast several neural network designs.

The findings show that the deep neural network's optimal configuration achieves a fraud detection rate of 91.79% while lowering the number of alerts by 35.16%. Since warnings identified by the neural network as false-positives would no longer require manual examination, the reduction in false-positive alerts has the potential to significantly lower the expenses associated with manual labour.

The proposed method holds the potential for automating fraud detection and advances the system in terms of the effectiveness of fraud detection systems by utilizing deep neural network capabilities. By lowering reliance on manual assessment of false-positive warnings, the findings emphasize the potential for cost savings and greater productivity. [11]

The paper emphasizes the need to identify financial fraud in the banking sector and recognizes the difficulties in doing so, including the need for interpretability and privacy legislation that restrict access to large-scale transaction data. To get beyond these restrictions, a lot of existing fraud detection techniques rely on manually created features. To address this, the authors suggest behaviour- and segmentation-based features that rely on statistical traits unique to fraudulent and non-fraudulent accounts. These characteristics offer clear cause-and-effect connections and show encouraging predictive outcomes.

The article also raises questions about the potential for unstable results when employing well-known boosting classifiers like XGBoost and LGBM because some features have time-inhomogeneous qualities. The authors used the Kolmogorov-Smirnov test to find and remove certain problematic characteristics to remedy this issue. As a result, XGBoost and LGBM classifiers have improved in terms of detection performance and resilience. According to experimental results, the suggested method performs better than competing classifiers like SVM and random forests.

The article also discusses the drawbacks of creating training and testing sets via random sampling since it misses time inhomogeneity and leads to inaccurate results of the accuracy of machine learning models. The

performance of resampling techniques used to address data imbalances in fraud detection is also impacted by the time-inhomogeneous traits found in fraud patterns. The authors find that due to the various modus operandi patterns, inappropriate linear interpolation in SMOTE-related techniques results in subpar performance. The essay contends that this problem can be solved by synthesizing false samples with generative adversarial networks (GANs) and straightforward oversampling. [12].

The article examines the rise in financial transactions and the associated rise in fraud instances, focusing in particular on credit card purchases made through online shopping sites. Due to the enormous expenses involved, finding fraudulent behavior in these transactions has become a top priority. The study uses Bayesian optimization to tune the involved hyperparameters while taking into account relevant factors like imbalanced data. For improving the efficiency of the LightGBM technique, it proposes weight-tuning as a pre-processing step for resolving data imbalance and offers a voting mechanism by merging CatBoost and XGBoost algorithms. For additional performance enhancement, deep learning techniques are used to fine-tune the hyperparameters with a focus on the suggested weight-tuning method.

The article's main focus is on the use of machine learning models to spot fraudulent transactions. Since the majority of systems often identify fraudulent acts after they have already occurred, it draws attention to how difficult it is to detect fraud in real-time or almost real-time. The highly unbalanced character of fraudulent transactions, which occur significantly less frequently than legitimate transactions, makes the task of fraud identification much more challenging.

The outcomes show that when applied to the highly unbalanced bank loan dataset, the quantum-enhanced support vector machine beats competing techniques in terms of speed as well as accuracy. However, its performance for Israel credit card transaction data is on par with other approaches. The study also shows that feature selection considerably increases detection speed while having a negligible effect on accuracy [13].

The article provides information on how to choose the best methods for a given dataset while taking into account the trade-offs between efficiency, precision, and expense in fraud detection activities. [14]

The article covers the issue of online fraud in e-commerce platforms and focuses on the value of reputation scores offered by platform users to assess vendors. To improve their earnings, sellers want to earn high reputation scores. But by combining it, scammers can misrepresent reputation scores and get unwary customers.

The study consists of conceptual propositions that refer to both people and indications of transactions and focus on fraud transaction attributes. To enhance the accuracy of fraud detection, two more independent variables – product type and product nature are added. Using a real-world undefined dataset, the effectiveness of these indications and the detection model is confirmed, enabling at least the separation between criminal and legitimate transactions

In conclusion, the research presented can be considered to contribute a usable and enriching conceptual frame-work concerning big data technologies and data mining approaches in an attempt to extract the relevant signs from fraud transactions in a bid to make the identification. These product attributes make fraud detection more accurate, and real-world data is deployed to evaluate the effectiveness of the recommended indicators and the model. [15]

The article is primarily devoted to the problem of class imbalance data classification, which has recently attracted the attention of researchers in science disciplines such as fraud detection, metabolomics, and cancer diagnosis. It stresses the undesirable consequences of sharing similarities in class-imbalanced learning performance, which needs to have methods that eliminate the overlaps and enhance the classification performance.

The paper suggests a model for feature selection that reduces data overlap and in turn increases classification ac-curacy. This method is based on enhanced R-value. These algorithms make use of sparse feature selection methods and, in the case of ROS and ROA, resample data. The algorithms were created primarily for binary classification jobs, it is important to note.

Results from simulations show how well the suggested techniques control the fluctuation in the false discovery rate when choosing the primary features for process modelling. Four credit card datasets are used in trials to assess the algorithms' performance. Evaluation measures like F-measure and G-mean demonstrate how much better the pro-posed algorithms are than conventional feature selection techniques.

To reduce overlap in class-imbalanced data classification, the study offers a feature selection technique based on enhanced R-value. The proposed approach can also be proved when testing on credit card datasets and showed higher accuracy than traditional feature selection methods. According to the study, the presented efficient feature se-lection method can also be applied to tackle overlapping issues in the context of other problems associated with class-imbalanced learning issues. [16]

Recall, precision, and accuracy are some metrics used to assess the models. The presented approach is further vali-dated with the use of a highly inclined synthetic credit card fraud dataset.

The result of experiments shows that using AdaBoost provides better results in terms of effectiveness. From the perspective of evaluation, the imposed alterations lead to increased effectiveness compared to the alternatives.

Hence, the study concludes with a machine learning-based credit card fraud detection framework. The AdaBoost algorithm is incorporated into the system to enhance the feature’s classification and the system is

evaluated with re-al-world imbalanced data sets. The results acquired in the experiments depicted effectiveness in comparison to other approaches regarding credit card fraud detection. [17]

The article's main topic is credit card fraud, with a focus on the skimming method that fraudsters use to obtain card information. The suggested process entails numerous steps. In the beginning, principal component analysis (PCA) and autoencoder extractors are used to extract discriminative features. Next, using the K-Means method, comparable fraudulent transactions are clustered. The approach locates possible merchants implicated in the skimming scheme through retrospective analysis of all transactions by locating matched merchants within the created clusters.

Even with the lack of prior knowledge regarding existing skimming spots, experiments on real card transactions show encouraging results for the suggested strategy. Seven out of the nine locations of compromise that the bank had previously identified and reported were found after applying the approach. [18]

The article highlights the difficulty of detecting credit card fraud in online purchases and emphasises the need for flexible fraud detection systems (FDS). It is challenging to adapt current methods to new problems, such as fresh payment systems, other countries, or particular population segments, due to the varied nature of fraud behaviour.

The challenge of transfer learning, which entails adjusting current detection models to new contexts, is the subject of this research.

The study makes use of a dataset from an industry partner that includes around 200 million transactions within six months. With an emphasis on accuracy across various transfer contexts, the essay describes and contrasts 15 transfer learning strategies, spanning from fundamental baselines to cutting-edge and unique approaches. Two key conclusion emerge from the evaluation: (i) the availability of labelled samples in the target domain significantly influences the effectiveness of transfer methods, and (ii) an ensemble solution based on self-supervised and semi-supervised domain adaptation classifiers is proposed to address this challenge.

From the findings of the experiments, it can be observed that the proposed ensemble solution yields lesser sensitivity to the hoe much of labelled samples available in the target domain and it performs with amazing accuracy in credit card fraud detection. [19]

As a result, to resolve the problem of imbalanced classification in credit card fraud detection the study proposes a state-of-art method in oversampling. This approach is based on Deep learning methods and variational automatic coding (VAE). To generate several instances from the minority, the VAE approach is applied. The classification net-work is then trained using such generated samples. The baseline classification model performs greatly and performance is enhanced by the extended dataset that is produced using the VAE method. Performance is measured in terms of F measure, precision, specificity and accuracy.

The findings of the work suggest the proposed VAE-based oversampling technique is highly effective in addressing the imbalanced classification problem. This approach enhances the performance of the classification model and enhances the capability of detecting fraud samples as it generates many synthetic samples from minority classes. [20]

III. MATERIALS AND METHODOLOGY

A. DATA COLLECTION AND PREPROCESSING

The dataset [21] used in this work is gathered from the well-known sight “Kaggle”. To address the unequal class distribution in the dataset, the SMOTE-ENN i:e edited nearest neighbor method and the hybrid synthetic methodology of minority oversampling are also used. This method outperforms various models that are commonly used by machine learning classifiers and mentioned in the literature review in terms of performance. the below figure.1 working of flowchart explains how data is collected step by step and further on for preprocessing and finalizing the accuracy achieved.

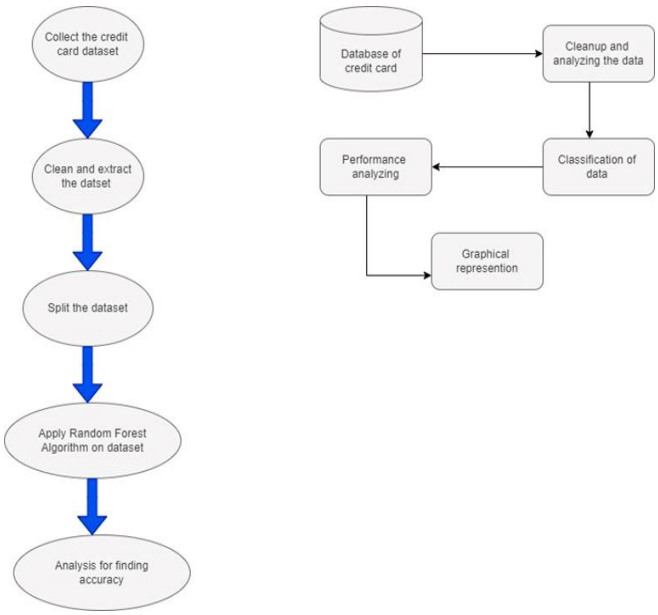

FIGURE 1: Data Collection and Preprocessing

B. DATA VISULIZATION

The dataset is plotted using the matplotlib library of Python was used. Following is the plot diagram of the respective dataset. To visually understand the dataset and look for discrepancies in it, we plot various graphs. Histogram, density plot, box plot, and scatterplot matrix. Dimensionality reduction is the following phase when we lower the number of dimensions in our data collection to enable visualization on a 2D or 3D display. Principal component analysis, or PCA, does this, so figure 2 display the class column which one is target showing fraud 1 and non-fraud 0.

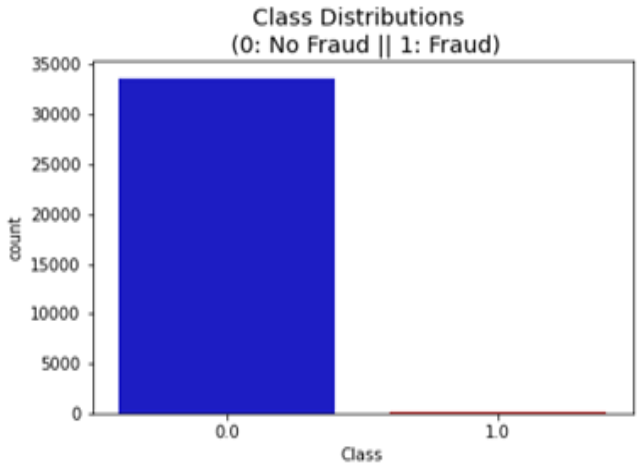

FIGURE 2: FIGURE OF FRAUDLUENT TRANSACTIONS

C. FEATURE SELECTION

To reduce overlap in class-imbalanced data classification, the study offers a feature selection technique based on enhanced R-value. Experiments on credit card datasets show that the suggested algorithms perform better than traditional feature selection techniques. According to the article, this efficient feature selection method can be used to tackle overlapping issues in other class-imbalanced learning issues.

D. RANDOM FOREST ALGORITHM

When compared to the client's prior exchanges, card exchanges are consistently new. It is recognized as an idea float issue in reality and this newness is a very difficult problem [1]. Idea float can be thought of as a variable that alters gradually and erratically over time. High levels of information irregularity are caused by these variables. The main goal of our investigation is to resolve the problem of Concept Float's inability to operate in certain circumstances. Table 1, [1] lists the fundamental components that are observed during any trade. Figure 3 demonstrate the workflow of credit card Transactions fraud history according to profile of customer and make it sure for final result.

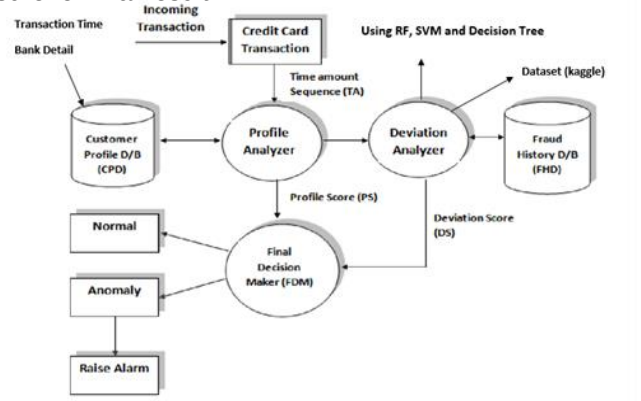

FIGURE 3: System Design for Credit Card Farud Detection using Random Forest classifiers

E. MODEL TRAINING AND EVALUATION

Separate the training and testing sets from the preprocessed dataset. Describe in detail how to use the training set to train a Random Forest model. Also, go over the hyperparameter tweaking procedure for model optimization. Discuss additional evaluation methods like cross-validation to ensure model effectiveness.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$MCC = \frac{TP * TN - FP * FN}{((TP + FP)(TP + FN)(TN + FP)(TN + FN))^2} \quad (3)$$

In the above equations parameters are used as:
TP represent as True Positive
TN represent as True Negative
FP represent as False Positive
FN represent as False Negative

IV. RESULTS

The algorithm puts out the number of false benefits it has identified and contrasts it with the true characteristics. This is used to determine the calculations' correctness and exactness score. 10% of the whole dataset was the little portion of data we used for quicker testing. Near the end, the entire dataset is also used, and both results are printed. These results, along with the arrangement report for each calculation, are provided in the result as follows, where class 0 denotes that the results are not certain to be significant and class 1 denotes that the method is not completely established as an extortion exchange. To look for false positives, this result was compared to the class values.

A. COMPARISON WITH MACHINE LEARNING ALGORITHMS

Table1 shows the comparison of the current model with some other machine learning models on which Credit Card Fraud detection had been performed previously

TABLE 1: PRECISION, ACCURACY, AND MCC VALUES

Method	Precision	Accuracy	MCC
SVM	0.782	0.997	0.5267
LR-Logistic	0.876	0.990	0.6786
Decision Tree classifier	0.884	0.944	0.8156
Random Forest Classifiers	0.9350	0.994	0.8368

In the Figure 4 The plot displays the ratio of Fraud and no fraud performance with the classifier model (2X2)

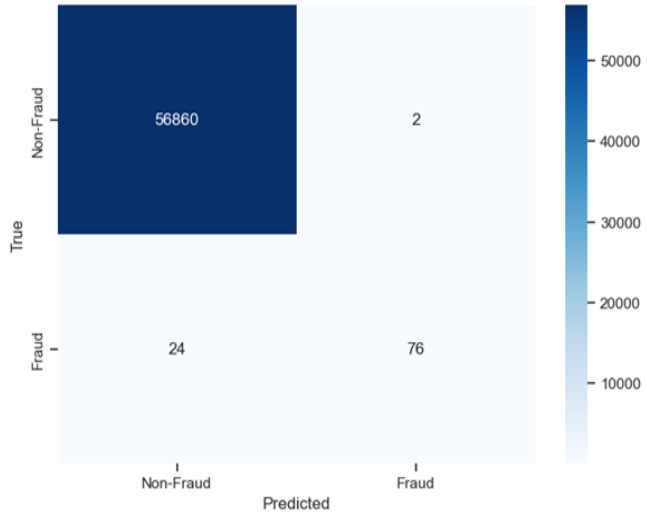

FIGURE 4: Confusion Matrix

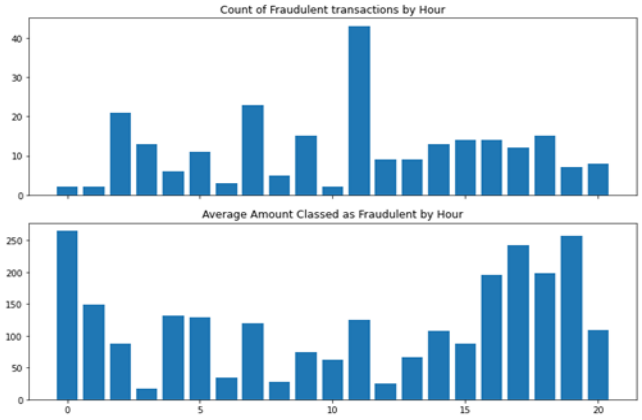

FIGURE 5: Histogram Plot

In the Figure 5 Given graph demonstrates the ratio of fraudulent transactions is a lot more than the original ones.

The aforementioned graphs in Figure 5, depict how fraudulent transactions were distributed throughout the days' time(hr) and hour banks.

Figure 6 display the Interpreting correlation with debit card transitions positive correlation and negative correlation.

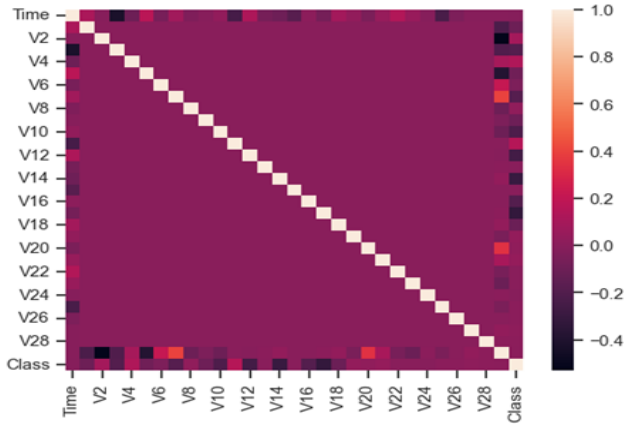

FIGURE 6: Interpreting correlation with Debit card Fraud

V. CONCLUSIONS

In this work we developed a new method for the detection of credit card fraud where users are gathered because of their transactions and focus on moral principle to develop a profile for each cardholder.

Charge card extortion is undoubtedly a sign of criminal unreliability. The most well-known extortion techniques, along with their methods of discovery, have been thoroughly examined in this article, which also looked at recent developments in the subject. Additionally, this study has shown in detail how AI may be used to improve extortion identification along with computation, pseudocode, clarification of its execution, and trial-and-error outcomes. Even if the computation has an accuracy of more than 99.6%, it is still only 28% accurate when only a tenth of the informational index is taken into account.

REFERENCES

[1] Luo, B., Zhang, Z., Wang, Q., Ke, A., Lu, S., & He, B. (2023). Ai-powered fraud detection in decentralized finance: A project life cycle perspective. ACM Computing Surveys.

[2] Bello, O. A., & Olufemi, K. (2024). Artificial intelligence in fraud prevention: Exploring techniques and applications challenges and opportunities. Computer Science & IT Research Journal, 5(6), 1505-1520.

[3] Mienye, I. D., & Jere, N. (2024). Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions. *IEEE Access*.

[4] Parkar, E., Gite, S., Mishra, S., Pradhan, B., & Alamri, A. (2024). Comparative study of deep learning explainability and causal ai for fraud detection. International Journal on Smart Sensing and Intelligent Systems, 17(1).

[5] Rezvani, S., & Wang, X. (2023). A broad review on class imbalance learning techniques. Applied Soft Computing, 143, 110415.

[6] Leevy, J. L., Hancock, J., & Khoshgoftaar, T. M. (2023). Comparative analysis of binary and one-class classification techniques for credit card fraud data. Journal of Big Data, 10(1), 118.

[7] Prabhakaran, N., & Nedunchelian, R. (2023). Oppositional Cat Swarm Optimization-Based Feature Selection Approach for Credit Card Fraud Detection. Computational Intelligence and Neuroscience, 2023(1), 2693022.

[8] Ralli, R., Jugran, G., Gaurav, M., & Goyal, M. (2024, August). An Ensemble based Fraudulent Blockchain Account Detection System. In Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing (pp. 337-342).

[9] Taha, A. A., & Malebary, S. J. (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE access, 8, 25579-25587.

[10] Khalid, A. R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., & Adejoh, J. (2024). Enhancing credit card fraud detection: an ensemble machine learning approach. Big Data and Cognitive Computing, 8(1), 6.

[11] Baria, J. B., Baria, V. D., Bhimla, S. Y., Prajapati, R., Rathva, M., & Patel, S. (2024). Deep Learning based Improved Strategy for Credit Card Fraud Detection using Linear Regression. *Journal of Electrical Systems*, *20*(10s), 1295-1301.

[12] Bao, Q., Wei, K., Xu, J., & Jiang, W. (2024). Application of Deep Learning in Financial Credit Card Fraud Detection. Journal of Economic Theory and Business Management, 1(2), 51-57.

[13] Du, H., Lv, L., Wang, H., & Guo, A. (2024). A novel method for detecting credit card fraud problems. Plos one, 19(3), e0294537.

[14] Zhu, K., Zhang, N., Ding, W., & Jiang, C. (2024). An Adaptive Heterogeneous Credit Card Fraud Detection Model Based on Deep Reinforcement Training Subset Selection. IEEE Transactions on Artificial Intelligence.

[15] Chatterjee, P., Das, D., & Rawat, D. B. (2024). Digital twin for credit card fraud detection: Opportunities, challenges, and

fraud detection advancements. Future Generation Computer Systems.
[16] Chhabra, R., Goswami, S., & Ranjan, R. K. (2024). A voting ensemble machine learning based credit card fraud detection using highly imbalance data. Multimedia Tools and Applications, 83(18), 54729-54753.
[17] Ning, W., Chen, S., Lei, S., & Liao, X. (2023). Amwspladaboost credit card fraud detection method based on enhanced base classifier diversity. IEEE Access.
[18] Salekshahrezaee, Z., Leevy, J. L., & Khoshgoftaar, T. M. (2023). The effect of feature extraction and data sampling on credit card fraud detection. Journal of Big Data, 10(1), 6.
[19] Jemai, J., Zarrad, A., & Daud, A. (2024). Identifying Fraudulent Credit Card Transactions using Ensemble Learning. IEEE Access.
[20] Zioviris, G., Kolomvatsos, K., & Stamoulis, G. (2024). An intelligent sequential fraud detection model based on deep learning. The Journal of Supercomputing, 80(10), 14824-14847.

Date available online: 23-07-2025
Vol. 3, Issue 1 (January – June 2025)

DOI: https://doi.org/10.24312/ucp-jeit.03.01.485

Maximum Efficiency Point Tracking Control for Dynamic Wireless Power Transfer

Muhammad Umer Noor[1], Muhammad Anique Aslam[2], Syed Abdul Rahman Kashif.[3]
[1]Department of Electrical Engineering, University of Engineering and Technology, Lahore 54890, Pakistan (e-mail: umernoor.75@gmail.com)
[2]Department of Electrical Engineering, University of Engineering and Technology, Lahore 54890, Pakistan (e-mail: maniqueaslam@uet.edu.pk)
[3]Department of Electrical Engineering, University of Engineering and Technology, Lahore 54890, Pakistan (e-mail: abdulrahman@uet.edu.pk)

Corresponding author: Muhammad Anique Aslam (e-mail: maniqueaslam@uet.edu.pk).

ABSTRACT

Wireless power transfer has evolved as a dominant research field. Roadway-powered vehicles have poor and fluctuating efficiency due to varying coupling coefficients because of their dynamic nature. Efficiency improvement efforts become useless in the event of communication failure between the transmitter and receiver sides. This paper presents a novel algorithm based on a silent handshake between the two sides to achieve the most efficient operation with varying flux linkage. The timed sequence of operations enables the continuous load mapping by impedance matching converter to maximize the efficiency followed by load regulation done by the supply converter based on accurate mathematical modelling. Simulation and experimental results are presented to ascertain the performance of the algorithm and the mathematical models.

INDEX TERMS Dynamic coupling, Impedance matching converter, LCC inverter, Maximum efficiency point tracking, Wireless power transfer

I. INTRODUCTION

WIRELESS power transfer has been an interesting domain of research for more than 100 years [1]. Various transfer methods are used such as microwave, laser, inductive and capacitive methods for transmitting power wirelessly, each one having its list of merits and demerits [2], [3]. The microwave power transfer occurs by the microwaves travelling from transmitter to receiver. This can be used over long distances but is not safe for human life [4]. Laser has also played a vital role in power transfer over long distances. However, its use is limited for communication purposes only because of significant attenuation caused by the medium of propagation [5]. The capacitive power transfer occurs through electrostatic induction transferring power via an electric field. This topology is extensively studied but finds little to no room in commercial applications [6]–[8]. Inductive power transfer works on electromagnetic induction transferring power via varying magnetic fields arising from an AC magnetic field [9]. This method has proved its dominance in commercial applications. Moreover, resonant inductive wireless power transfer further amplifies the utility by enabling the power transfer at resonance which improves the transfer capability and efficiency [10].

In inductive coupling, power is transferred from transmitter coils embedded in road structure to receiver coils fitted in the vehicle based on electromagnetic induction [11]–[13]. A relative motion between the transmitter and the receiver coil changes the coupling causing the power transfer efficiency to drop at low coupling values which happens at reduced physical overlapping. Employing a mechanism that can dynamically track maximum efficiency points can substantially improve the energy transfer with continuously varying coupling coefficient [5], [6], [9], [10]. Various configurations of transmitter and receiver coils that provide better coupling parameters under misalignment conditions have been presented in [14], [15]. Some researchers have considered an array of transmitters powered by individual sources along the length of the road [16], [17]. Those coils are separated by an air gap to model the practical scenario since the air gap in the case of Electric Vehicles (EVs) ranges from 100 to 300 mm [9]. In addition, the effect of cross-couplings between single receivers having an overlapping over two transmitters are examined in [15]–[18] which happens at the boundary conditions in which the receiver is entering from one transmitter region to the next. A more complex version comes into play when there are multiple receivers over an array of transmitters which is a more realistic assumption since there can be an undetermined number of EVs on a single road. This leads to cross couplings within one receiver over two transmitters and within two or more receivers as well. In addition to making the above mentioned situations practically possible, there comes another idea of making this all happen at maximum efficiency while maintaining the load requirements. Multiple algorithms are presented in [9], [10] which can achieve the desired outputs. Some of the efficiency tracking algorithms are based on the Perturb and Observe (P&O) approach which slowly yields to maximum efficiency point [19], [20]. In the P&O based approach, the operating parameters of the circuit are adjusted with each iteration and efficiency is monitored. If it increases or decreases, it means the direction of propagation is right or wrong respectively. Numerous iterations are needed to reach the most efficient point because of small increments or decrements in operating points. Such an approach can have variable convergence time in dynamic power transfer conditions. In addition, since the efficiency plot of an experimental setup can have two or more peaks, the P&O based algorithms can track an efficiency point that is less efficient than its operating

point due to its working principle. Some researchers have employed a communication based approach as well to make the algorithms faster and obtain desired outputs in less time [16], [19], [21]. These options although have relieved the slower convergence rates but come with inherent deficiency due to their dependence on the communication channel. Any disruption in accurate information transfer between the receiver and transmitter can render the whole system of wireless power transfer entirely useless. Such a failure is unaffordable in real-world applications. Another promising control aspect is considered in [20] which maximizes system efficiency by controlling converter voltage ratio with load variation. But [20] does not consider the dynamic nature of flux linkage arising from the relative motion of the coils. Authors of [22] have deployed a bi-directional coupler with a cascaded buck and boost converter to yield maximum efficiency but this is based on the P&O approach and also utilizes a communication channel between transmitter and receiver sides. Another method of maximizing efficiency employed by researchers is the use of numerical techniques to estimate coupling coefficient [18]. This technique generally does not require a dedicated communication linkage. Circuit parameters are calculated by numerical methods and then operation point adjustments are done to ensure efficiency improvement. This way of tackling the problem is good but the performance resorts to the operational performance of the technique being used. For example in [18], they have considered a segmented array of transmitter coils in which the coupling coefficient is estimated with the recursive least square method followed by maximum efficiency point tracking but it requires a dedicated digital signal processor to do all the numerical computations at a high rate. This makes the design complex and implementation costly. Therefore, a more dependable algorithm that is not slow like P&O and also independent of communication requirements, needs to be presented that can yield the desired requirements of load regulation while maintaining the most efficient operation. Such a method should also be easily extendable to more complex scenarios discussed above.

From the above discussion it is evident that the majority of researchers have used a communication-based scheme to transfer parameter information including load voltage and current, duties and output voltage of converters between transmitter and receiver controllers to achieve the desired goals. These parameters are required to perform mathematical computations by controllers of each side so that the Maximum Efficient Point Tracking (MEPT) is achieved with load regulation. Keeping such limitations in view, this paper presents the estimation of the coupling coefficient with an inductive wireless power transfer model-based novel algorithm that can achieve maximum efficiency point operation without communication. Major contributions of this research are as follows.

- Accurate mathematical modelling of resonant inductive wireless power transfer
- Dynamic estimation of the coupling coefficient
- Continuous tracking of maximum efficiency point
- Enhancement in overall power transfer capability
- Eradication of communication link between the transmitter and receiver, thus, providing autonomous control on both sides
- Stable operation under low coupling conditions
- Reduction of battery bank size and eventually the cost of EVs
- Solving the range anxiety problem hindering the vast utility of EVs

This paper is organized as follows. Section II discusses the development of mathematical models and the designing of an algorithm for maximizing efficiency in a single transmitter- receiver system. Simulation and hardware prototyping of the experimental layout of the implemented wireless power transfer system along with outcomes is discussed in Section III. Finally, conclusions and prospects are discussed in Section IV.

II. MATHEMATICAL FRAMEWORK AND PROPOSED MEPT ALGORITHM

A single transmitter-receiver topology is considered with a series resonant receiver link to avoid any voltage drop. An LCC network is utilized because of independent transmitter coil current for the load [6] and its DC input property [16] which helps in establishing a silent handshake in this particular work [21], [23]. Conventionally source end converter helps in MEPT and the load end converter ensures voltage regulation. A very promising result is obtained as the roles are switched between the two sides. Fig. 1 shows the block diagram of the transmitter-receiver system. The following sections discuss the mathematical modelling and the proposed MEPT algorithm.

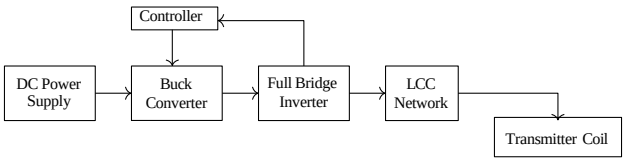

FIGURE 1. Block diagram of the transmitter-receiver system under consideration

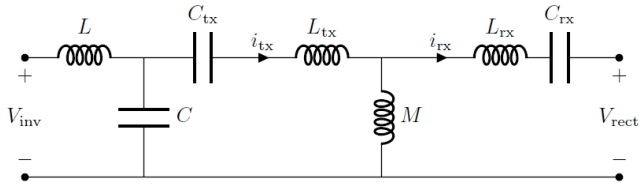

FIGURE 2. Equivalent lumped circuit of the proposed wireless power transmission system

A. LCC NETWORK

The LCC network which is fed by the inverter, as shown in Fig. 1, has a transmitter coil current given by the loop equation (1),

$$I_{tx}\left[\frac{1}{j\omega C_{tx}}+\frac{1}{j\omega C}+j\omega L_{tx}\right]=\frac{I_1}{j\omega C} \qquad (1)$$

where, I_{tx} is transmitter coil current, ω is angular switching frequency, L and C are LCC resonant inductance and capacitance respectively and V_{inv} is DC voltage at inverter input. Here, L and C are in

resonance to maintain the DC input property of the LCC inverter and ensure the independence of I_{tx} on receiver side circuit parameters. Therefore, using $\omega = 1/\sqrt{(LC)}$ to manipulate Equation (1) gives I_{tx} according to Equation (2).

$$I_{tx} = jC\omega V_{inv} \quad (2)$$

B. RECEIVER LC NETWORK

The receiver LC series network is also resonant to avoid any voltage drop. Consequently, the full voltage induced in the receiver coil appears at the rectifier input. The equivalent circuit of the transmitter and receiver coil linked with mutual inductance is shown in Fig. 2. The loop equations from Fig. 2 can be written as,

$$I_{rx}\left[\frac{1}{j\omega C_{rx}} + j\omega M + j\omega L_{rx}\right] - I_{tx}[j\omega M] + V_{rect} = 0 \quad (3)$$

where, I_{rx} is receiver coil current, M is mutual inductance of transmitter and receiver coils, L_{rx} and C_{rx} are receiver resonant inductance and capacitance respectively and V_{rect} is rectifier output voltage. Since, the receiver circuit is also resonant, so, putting $\omega = 1/\sqrt{(L_{rx}C_{rx})}$ in Equation (3) and using the fact that $I_{rx} = 0$ when receiver circuit is open, we get the magnitude of the rectifier voltage as given in Equation (4).

$$V_{rect} = j\omega M I_{tx} \quad (4)$$

C. COUPLING COEFFICIENT AT THE RECEIVER SIDE

The coupling coefficient (k) is an essential constituent in MEPT as it is required to track the optimum load. It quantifies the extent of flux linkage between the two coils. When the transmitter and receiver coils are in relative motion, flux linkage varies which changes k. This change needs to be tracked to accomplish MEPT.

Since mutual inductance between the two coils is related to their self-inductances and coupling coefficient, using $M = k\sqrt{(L_{tx}L_{rx})}$, Equation (2) and Equation (4) we get the coupling coefficient at the receiver side as given in Equation (5).

$$k = \frac{V_{rect}}{V_{inv}\sqrt{L_{tx}L_{rx}}\omega^2} \quad (5)$$

It should be noted that with the help of the startup cycle, V_{inv} is known by the receiver controller, ω is the switching frequency, L_{tx} and L_{rx} are coil parameters and V_{rect} is measured by the receiver controller. Consequently, k can be easily computed at the receiver side.

D. COUPLING COEFFICIENT AT THE RECEIVER SIDE

Inherently constant voltage operation at the load end is utilized by the receiver side controller which alters the Thevenin equivalent impedance seen by the transmitter side due to varying coupling causing the operation to deviate from the maximum efficiency point. Conventionally, the receiver side controls the load requirements while the transmitter side ensures efficient operation. Simultaneously maintaining constant load voltage and ensuring maximum efficiency is not possible just by one DC converter. The impedance transformation ratio of the buck-boost converter gives the optimum duty cycle of the converter for mapping load to the optimum value as,

$$D_{opt} = \frac{\sqrt{\frac{R_L}{R_{Lopt}}}}{1 + \sqrt{\frac{R_L}{R_{Lopt}}}} \quad (6)$$

where, D_{opt} is the duty of the buck-boost converter which maps the load R_L to the optimum load R_{Lopt}. At this point of operation, MEPT is achieved by the mapping of load to optimum value and transmitter controller action begins where the first step is the calculation of inverter input resistance which is given in Section II-E.

E. DUTY CYCLE FOR OPTIMAL LOAD TRACKING

The input resistance of the LCC inverter can be easily measured by using DC values. This can be done by sensing voltage and current at the output of the buck converter due to the DC input nature of the LCC inverter using,

$$R_{inv,in} = \frac{V_{inv,in}}{I_{inv,in}} \quad (7)$$

where, $V_{inv,in}$ and $I_{inv,in}$ are DC in nature and can be readily measured with potential divider and hall effect current sensor together with ADC of transmitter side controller as explained in Section III.

F. REFLECTED RESISTANCE

Reflected resistance can be easily computed from the inverter input resistance as [12],

$$R_{refl} = \frac{\frac{\pi^2}{8}}{\left[R_{inv,in} - \frac{\pi^2}{8}\right]\omega^2 C^2} - R_{tx} \quad (8)$$

where, R_{refl} is the reflected resistance as seen by transmitter side, $R_{inv,in}$ is the inverter input resistance and R_{tx} is the parasitic resistance of the transmitter coil.

G. INVERTER INPUT RESISTANCE

Reflected resistance in resonant wireless power transfer after the MEPT is done appears as shown in Equation (9).

$$R_{refl} = \frac{\omega^2 M^2}{\frac{8}{\pi^2}R_{Lopt} + R_{rx}} \quad (9)$$

Putting $M = k\sqrt{(L_{tx}L_{rx})}$ and $R_{Lopt} = (8/\pi^2)\, kR_{rx}\sqrt{(\omega^2 L_{tx}L_{rx}/R_{tx}R_{rx})}$ from Equation (22) derived in Section II-J gives Equation (10) for k at the transmitter side.

$$k = \frac{\frac{64}{\pi^4}R_{rx}R_{refl}}{\omega L_{tx}L_{rx}}\sqrt{\frac{L_{tx}L_{rx}}{R_{tx}R_{rx}}} \quad (10)$$

The knowledge of the coupling coefficient value gives complete information about the receiver parameters, load and duty cycle of the load converter which helps the transmitter controller in load regulation.

H. INVERTER INPUT RESISTANCE

Since the transmitter coil current depends on DC voltage input to the inverter, this voltage can be controlled by a DCDC converter which takes feed from DC power supply as shown in Fig.1. Load voltage (V_L) is related to V_{rect} by

buck boost voltage transformation ratio as,

$$V_{\text{rect}} = \frac{1-D}{D} V_L \qquad (11)$$

where, D is the load end converter duty cycle. Substituting V_{rect} from Equation (4) and I_{tx} from Equation (2) in Equation (11) gives Equation (12).

$$V_{\text{inv}} = \frac{\frac{1-D}{D} V_L}{\omega^2 MC} \qquad (12)$$

It should be noted that the duty cycle of the load end converter is also estimated by the transmitter controller to calculate the inverter input voltage necessary for load regulation. Since k is already estimated, this duty cycle value is easily computed by the same expressions as derived for the receiver side controller. The transmitter controller does all the computations to reach Equation (12). Maintaining this voltage at the inverter input ensures load voltage regulation, thus, completing the silent handshake between the transmitter and the receiver.

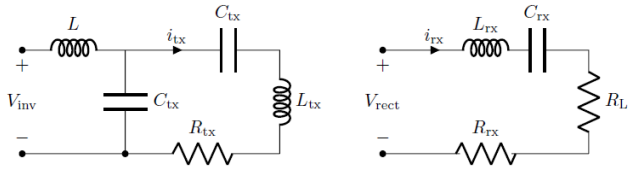

FIGURE 3. Equivalent transmitter and receiver circuits of the wireless power transfer system

I. INVERTER INPUT RESISTANCE

Once the receiver controller is done with MEPT, the transmitter controller can easily estimate the load value based on optimum load, coupling coefficient, load converter duty cycle and reflected resistance. Using impedance transformation relation of the load converter, R_{Lopt} can be calculated according to Equation (13).

$$R_{\text{Lopt}} = \frac{(1-D)^2}{D^2} R_{\text{Ltx}} \qquad (13)$$

Ignoring the parasitic resistance of the receiver coil and substituting Equation (13) in Equation (8) results in Equation (14).

$$R_{\text{Ltx}} = \frac{\omega^2 M^2}{\frac{8}{\pi^2} R_{\text{refl}} \left(\frac{1-D}{D}\right)^2} \qquad (14)$$

J. REFLECTED RESISTANCE

There exists a certain load value that maximizes the power transmission efficiency which is obtained by partial derivation of efficiency expression with respect to the load. Fig. 3 shows the primary and secondary equivalent circuits of the wireless power transfer system used in this paper where R_{tx} and R_{rx} are winding resistances of the transmitter and receiver coils. Since the receiver circuit is resonant, so, full induced voltage appears across the rectifier input after the drop across the winding resistance. From Fig. 3, load power is given by Equation (15).

$$P_{\text{L}} = I_{\text{rx}}^2 R_{\text{L}} \qquad (15)$$

Since the receiver circuit is resonant, so, I_{rx} is given by Equation (16).

$$I_{\text{rx}} = \frac{V_{\text{rect}}}{R_{\text{L}} + R_{\text{rx}}} \qquad (16)$$

Manipulation of Equation (15) and Equation (16) gives PL as given by Equation (17).

$$P_{\text{L}} = \frac{I_{\text{tx}}^2 \omega^2 M^2 R_{\text{L}}}{(R_{\text{L}} + R_{\text{rx}})^2} \qquad (17)$$

Ignoring switching losses, input power is given by Equation (18),

$$P_{\text{in}} = I_{\text{tx}}^2 R_{\text{tx}} + P_{\text{rx}} \qquad (18)$$

where, Prx is the total receiver side power which is given by Equation (19).

$$P_{\text{rx}} = I_{\text{rx}}^2 (R_{\text{rx}} + R_{\text{L}}) \qquad (19)$$

Power efficiency is given by Equation (20),

$$\eta = \frac{P_{\text{L}}}{P_{\text{in}}} \qquad (20)$$

where, P_L is the load power and P_{in} is the source input power. Equation (18), Equation (17), Equation (4) and Equation (16) give η according to Equation (21).

$$\eta = \frac{\omega^2 M^2 R_{\text{L}}}{R_{\text{tx}}(R_{\text{L}} + R_{\text{rx}})^2 + \omega^2 M^2 (R_{\text{L}} + R_{\text{rx}})} \qquad (21)$$

In order to find the optimal load that maximizes the efficiency, partial differentiation of the efficiency expression given in Equation (21) is performed with respect to RL which gives an optimum value of RL according to Equation (22).

$$R_{\text{L}} = R_{\text{rx}} k \sqrt{\frac{\omega^2 L_{\text{tx}} L_{\text{rx}}}{R_{\text{tx}} R_{\text{rx}}}} \qquad (22)$$

This expression shows that once k is known, the optimal load can be easily computed based on circuit element values. After the calculation of the optimum load value for a given coupling coefficient, the load end converter calculates the optimum duty cycle required for the said mapping. This duty cycle is fed by the receiver side controller, thus, terminating the receiver side action. At this stage, the MEPT is completed and load regulation is then achieved by the transmitter side controller.

Because of the absence of communication between both sides, the transmitter controller can only estimate the receiver side parameters based on the reflected resistance value. Here, the DC input property of the LCC inverter plays a helpful role. By measuring the DC parameters at the input of the inverter, reflected resistance can be easily calculated. This also eases the practical bottleneck of measuring high frequency AC parameters which requires sophisticated sensors and measurement setups. On the contrary, the DC domain measurement is easily done by analogue to digital converters available in embedded controllers and

potential dividers.

This reflected resistance calculation triggers the transmitter controller to take action after calculating k which completes the novel handshake enabling the complete knowledge of the receiver side parameters without any feedback channel. Therefore, load regulation is achieved by the transmitter controller.

K. REFLECTED RESISTANCE

Since this proposed scheme works by eradicating communication links fundamentally used in major research works, a start-up cycle is needed. In such a cycle, the transmitter and receiver controllers maintain the startup parameters like voltages and equivalent resistance. These parameters are known by both side controllers and help start the MEPT process.

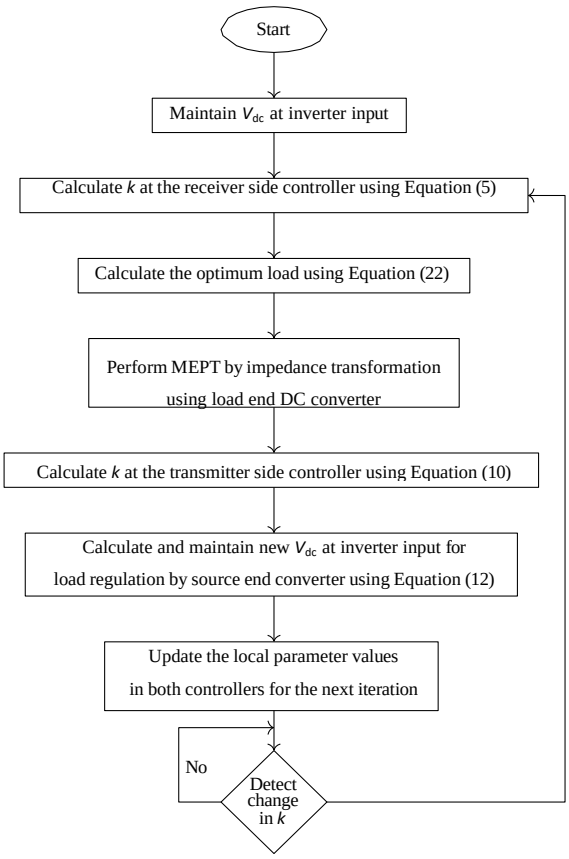

FIGURE 4. Flow chart of the proposed MEPT algorithm

Once the startup cycle is completed and MEPT is achieved for a given value of coupling coefficient, the whole system moves to a new equilibrium state which is remembered by each side controller. Further continuous changes in k are based on the memory function to achieve MEPT. This is further illustrated by the algorithm presented in Fig. 4. As indicated in Fig. 4, the algorithm starts by maintaining Vdc at the inverter input by means of the inverter side controller. After that, the coupling coefficient at the receiver side is calculated using Equation (5) of the mathematical model and the optimum load value is calculated using Equation (22). This is followed by MEPT performance by the load end DC converter using impedance transformation. After the completion of the MEPT at the load side, the coupling coefficient at the transmitter side is calculated using Equation (10). Consequently, the source end converter performs load regulation by calculating and maintaining the new Vdc at the inverter input using Equation (12). After that, the local parameter values are updated in both the receiver and transmitter side controllers for the next iteration that starts when the change in coupling coefficient is detected.

After the load regulation is achieved for a given state, both controllers now come to a rest state and wait for the next change in k to restart the MEPT algorithm which eventually reaches a new equilibrium point.

III. RESULTS AND DISCUSSION

This section discusses simulation and experimental results to ascertain the performance of the algorithm and the mathematical models.

A. SIMULATIONS

The experimental layout is first simulated in PSIM software to check the accuracy of mathematical models, theoretical analysis and working of the MEPT algorithm. The LCC inverter board is fed by a buck converter which takes input from a DC source set at 100 V. The LC series network at the receiver side is kept resonant at a switching frequency of 100 kHz which feeds a full bridge rectifier. The rectifier is connected to a buck-boost converter which feeds an electronic load. The charging of the batteries is emulated by the DC electronic load. More details on battery charging methods in EVs can be found in [24], [25], [26] and [27].

Promising results are obtained from the simulation data. Simulations are performed for a load power of 90 W when the required load voltage is set to 30 V. The total simulation time is 85 ms and the run time is around 8 minutes on a laptop having an Intel Core i3 4010 CPU running at 1.7 GHz with 4 GB RAM.

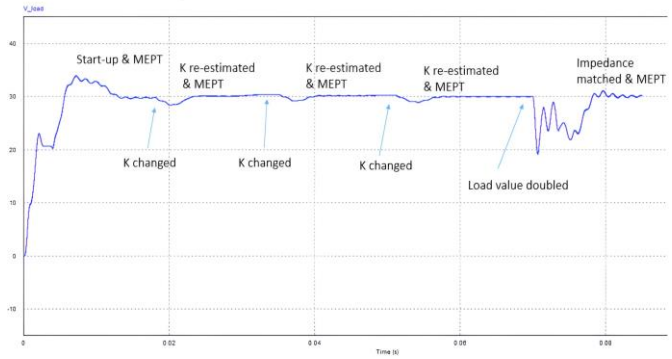

FIGURE 5. Load voltage profile representing load voltage with respect to time under varying coupling conditions. This figure indicates the variation in the transmitter and receiver coupling due to the relative motion between the two which leads to variations in the coupling coefficient. These changes in the coupling coefficient trigger the proposed MEPT algorithm that re-estimates the coupling coefficient and, hence, a steady state point is achieved as indicated by the marked points.

Figure 5 shows the waveforms to depict the working of the proposed MEPT algorithm. From 0 to around 18 ms, the start-up cycle executes and the system attains MEPT. At around 18 ms, *k* is reduced and the system goes into *k* reestimation followed by MEPT attainment at around 32 ms. At 35 ms, *k* is reduced and the algorithm is executed again to achieve MEPT at 50 ms and 70 ms. At around 70 ms, the load value is halved, which again disturbs the

system. Since *k* remains the same at this instant, the receiver side controller maps this load to the same old optimal load and the transmitter controller readjusts its voltage based on the load converter's updated duty cycle value to maintain the load voltage.

TABLE 1. Values of various parameters and circuit components of the hardware prototype measured at 100 kHz

Components/Parameters	Value/s
WPT Link	
L_{tx}/R_{tx}	34.38uH/0.1614Ω
L_{rx}/R_{rx}	45.936uH/0.0757Ω
Switching Frequency (f_s/w)	100kHz/628.32kRad/s
Coils Separation	10 cm
Coupling Coefficient Range	0.07 – 0.3
Micro-Controller	STM32f407
Inverter Components	
MOSFET	SCT-3030ALGC11-ND
Gate Driver	SIC8274
Rectifier	
Diode	NTST30100
DC Converters	
MOSFET	IPP114N12N3GXKSA1
Gate Driver	SIC8271

B. HARDWARE PROTOTYPE

A hardware prototype of the experimental layout is built to ascertain the theoretical and simulation results. Circuit specifications and details are shared in Table 1. Lab grade DC supply feeds the buck converter which maintains a constant voltage. Separate microcontrollers control each transmitter and receiver side circuit. STM 32F407 microcontroller operates these converters by using PI control. The transmitter and receiver coils are made on acrylic plastic sheets using litz wire. The distance between these is kept at 10 cm using cardboard. In experimentation, the load end converter is changed to boost from buck-boost type because practically the voltage at the output of the rectifier is always below 30V. The numerical commutations are performed by the STM board which is the heart of the hardware prototype. The duty cycle necessary for matching the load impedance to the optimum value is fed to the boost converter by the STM board to achieve MEPT. DC electronic load is kept at 10 Ω and is connected to the output of the boost converter. Fig. 6 shows the hardware prototype of the experimental layout.

FIGURE 6. Hardware setup of the experimental layout. 1. Receiver coil. 2. Transmitter coil. 3. Full bridge rectifier with receiver side resonant capacitor. 4. LCC compensation network. 5. Source end converter (buck type). 6.Transmitter side controller. 7. Full bridge inverter. 8. Load end converter. 9. Receiver side controller.

The transmitter circuit is kept stationary and the receiver side controller slides over it which causes the coupling coefficient to follow a parabolic path starting from a low overlapping with the coupling coefficient as low as 0.07 to the highest value of 0.3 as it catches the maximum flux when the centre of both the coils are aligned together. The duty cycle is kept slightly below 50% for body diodes to conduct the commutating current.

The main constituents of the experimental layout are first tested individually to meet their desired goals. The step response of the converters and inverter along with the wireless power transfer structure is checked to ensure linear convergence and stable operation. Fig. 7 shows the step response of the buck converter. The output of the converter is bounded and stable. Transient time is within 3 ms which does not pose any issue to algorithm convergence time. Similarly, Fig. 8 shows the step response of the boost converter which is also well-bounded and transient time is also adequate. Fig. 9 shows the step response of the full bridge inverter and rectifier along with the transmitter and receiver coils. The DC source is connected directly with the inverter and step input is introduced. The output waveform is captured at the rectifier output. In this way, the step response of the whole circuit is checked for its transient operation.

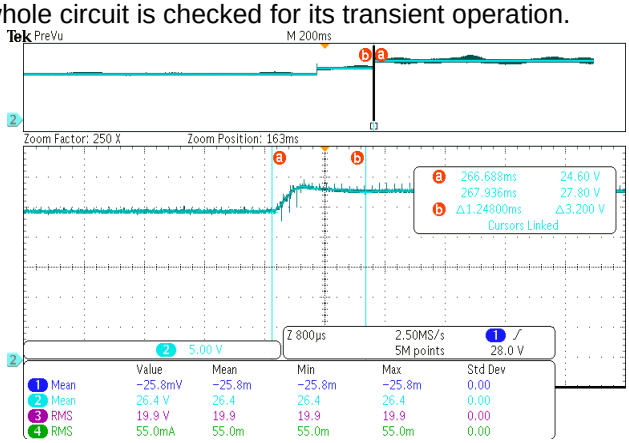

FIGURE 7. Step response of the buck converter representing a satisfactory transient response and a stable steady state response

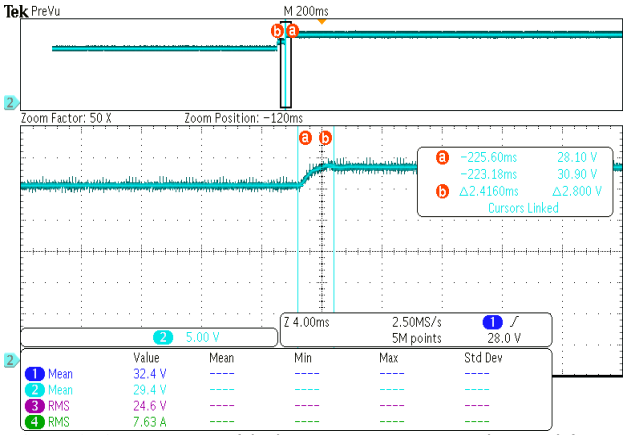

FIGURE 8. Step response of the boost converter representing a satisfactory transient response and a stable steady state response

Fig. 10 and Fig. 11 show the load voltage waveforms of the implemented wireless power transfer system in light blue.

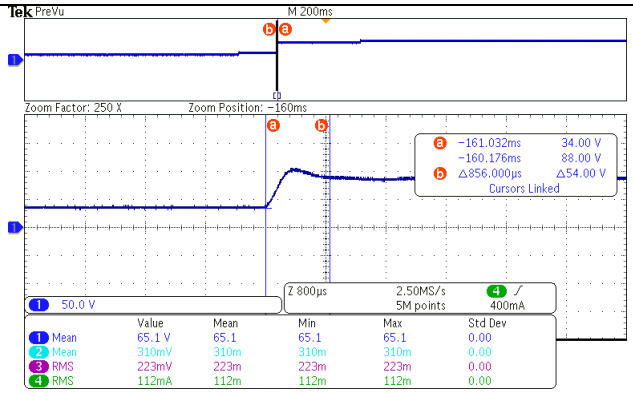

FIGURE 9. Step response of the rectifier and inverter representing a satisfactory transient response and a stable steady state response

This shows that there are variations when the flux linkage changes which are quickly maintained at the required value after MEPT. Algorithm convergence time is around 300 ms. Fig. 12 shows the actual and calculated values of the coupling coefficient. This shows that the estimated and actual values are in great coherence. Finally, Fig. 13 shows the power efficiency with and without the proposed MEPT algorithm. Significant efficiency improvement is observed which highlights the performance of the presented novel algorithm.

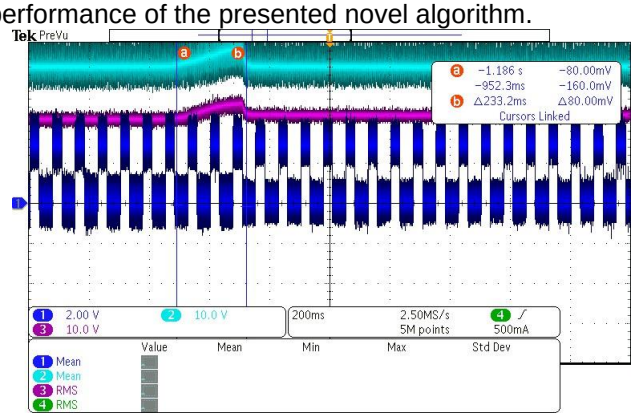

FIGURE 10. Load voltage profile for the rising coupling coefficient. The load voltage rises between points 'a' and 'b' as a result of an increase in the flux linkage due to an increase in the coupling coefficient caused by the increased overlapping between the transmitter and the receiver coils. This change in the coupling coefficient is detected and the voltage is re-established by the proposed MEPT algorithm.

IV. CONCLUSIONS AND FUTURE EXTENSION

This paper proposes a maximum efficiency point tracking algorithm that ensures load regulation criterion for dynamic wireless power transfer without the need for a communication medium between the transmitter and receiver side controllers or the use of numerical methods based on complex algorithms requiring higher processing capabilities by developing more accurate mathematical models of the inductive link circuit while benefitting from LCC inverter advantages. Careful theoretical analysis is done to yield a mathematical framework which paves the way for simulation and hardware prototyping. Simulation and experimental results are also presented which support the authenticity of the developed models and the proposed maximum efficiency point tracking algorithm. A technique based on a timed sequence of operations is developed which instantly reaches maximum efficiency by giving an exact optimum impedance which is matched by a DC converter instead of slowly converging towards it as presented in various other research works.

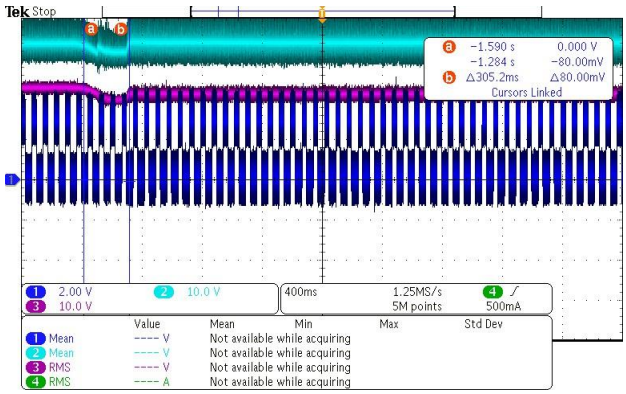

FIGURE 11. Load voltage profile for the falling coupling coefficient The load voltage decreases between points 'a' and 'b' as a result of a decrease in the flux linkage due to a decrease in the coupling coefficient caused by the reduced overlapping between the transmitter and the receiver coils. This change in the coupling coefficient is detected and the voltage is reestablished by the proposed MEPT algorithm.

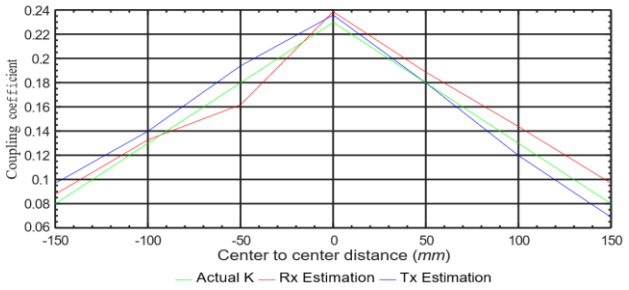

FIGURE 12. Coupling coefficient as estimated by controllers

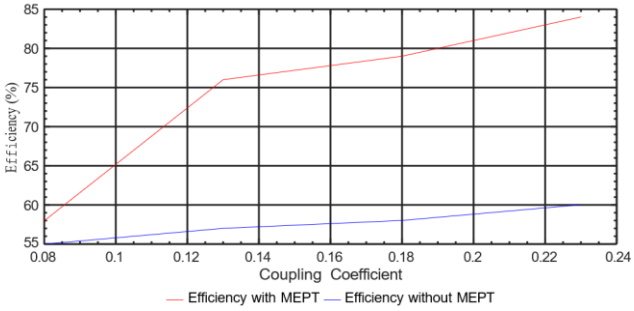

FIGURE 13. Efficiency plot with and without the proposed MEPT algorithm

This research work opens the possibility of expanding this approach to dynamic wireless power transfer at higher relative velocity. Significant efficiency improvement is observed with the proposed algorithm which confirms the vitality of the proposed study to be adopted for future extensions to more complex scenarios of multiple transmitter-receiver systems while also considering the effect of cross-couplings. Efforts can also be made to reduce the algorithm convergence time by improving the transient response of the subcircuits.

ACKNOWLEDGMENT

The authors wish to extend their gratitude for the support rendered by Ahn Dukju from Incheon National University, South Korea.

REFERENCES

[1] R. J., N. R, P. Vishnuram, C. Balaji, T. Gono, T. Dockal, R. Gono, and P. Krejci, "A review on resonant inductive coupling pad design for wireless electric vehicle charging application," Energy Reports, vol. 10, pp. 2047– 2079, 2023.

[2] Z. Bi, T. Kan, C. C. Mi, Y. Zhang, Z. Zhao, and G. A. Keoleian, "A review of wireless power transfer for electric vehicles: Prospects to enhance sustainable mobility," Applied Energy, vol. 179, pp. 413–425, 2016.

[3] J. Rahulkumar., R. Narayanamoorthi., P. Vishnuram, M. Bajaj, V. Blazek, L. Prokop, and S. Misak, "An empirical survey on wireless inductive power pad and resonant magnetic field coupling for in-motion ev charging system," IEEE Access, vol. 11, pp. 4660–4693, 2023.

[4] X. Zhu, K. Jin, Q. Hui, W. Gong, and D. Mao, "Long-range wireless microwave power transmission: A review of recent progress," IEEE Journal of Emerging and Selected Topics in Power Electronics, vol. 9, no. 4, pp. 4932–4946, 2021.

[5] Q. Zhang, W. Fang, Q. Liu, J. Wu, P. Xia, and L. Yang, "Distributed laser charging: A wireless power transfer approach," IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3853–3864, 2018.

[6] Z. Wang, Y. Zhang, X. He, B. Luo, and R. Mai, "Research and application of capacitive power transfer system: A review," Electronics, vol. 11, no. 7, 2022.

[7] S. Gillani, K. Shahid, M. Gulzar, and D. Arif, "Remaining useful life prediction of super-capacitors in electric vehicles using neural networks," Arabian Journal for Science and Engineering, vol. 49, pp. 7327–7340, May 2024.

[8] Rimsha, S. Murawwat, M. M. Gulzar, A. Alzahrani, G. Hafeez, F. A. Khan, and A. M. Abed, "State of charge estimation and error analysis of lithium-ion batteries for electric vehicles using kalman filter and deep neural network," Journal of Energy Storage, vol. 72, p. 108039, 2023.

[9] G. A. Covic and J. T. Boys, "Inductive power transfer," Proceedings of the IEEE, vol. 101, no. 6, pp. 1276–1289, 2013.

[10] I.-G. Sîrbu and L. Mandache, "Comparative analysis of different topologies for wireless power transfer systems," in 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe, pp. 1–6, 2017.

[11] Z. Bi, L. Song, R. De Kleine, C. C. Mi, and G. A. Keoleian, "Plug-in vs. wireless charging: Life cycle energy and greenhouse gas emissions for an electric bus system," Applied Energy, vol. 146, pp. 11–19, 2015.

[12] R. J, N. R, B. C, and S. A, "A dual receiver and inherent cc-cv operated wript ev charging system with high misalignment tolerance couplers," in 2023 IEEE International Transportation Electrification Conference (ITECIndia), pp. 1–8, 2023.

[13] R. J and N. R, "Power control and efficiency enhancement topology for dual receiver wireless power transfer ev quasi-dynamic charging," in 2023 IEEE International Transportation Electrification Conference (ITECIndia), pp. 1–6, 2023.

[14] S. Aldhaher, P. C.-K. Luk, and J. F. Whidborne, "Electronic tuning of misaligned coils in wireless power transfer systems," IEEE Transactions on Power Electronics, vol. 29, no. 11, pp. 5975–5982, 2014.

[15] C. Zhang, D. Lin, and S. Y. R. Hu, "Efficiency optimization method of inductive coupling wireless power transfer system with multiple transmitters and single receiver," in 2016 IEEE Energy Conversion Congress and Exposition (ECCE), pp. 1–6, 2016.

[16] D.-H. Kim, S. Kim, S.-W. Kim, J. Moon, I. Cho, and D. Ahn, "Coupling extraction and maximum efficiency tracking for multiple concurrent transmitters in dynamic wireless charging," IEEE Transactions on Power Electronics, vol. 35, no. 8, pp. 7853–7862, 2020.

[17] L. Shuguang, Y. Zhenxing, and L. Wenbin, "Electric vehicle dynamic wireless charging technology based on multi-parallel primary coils," in 2018 IEEE International Conference on Electronics and Communication Engineering (ICECE), pp. 120–124, 2018.

[18] D. Kobayashi, T. Imura, and Y. Hori, "Real-time coupling coefficient estimation and maximum efficiency control on dynamic wireless power transfer for electric vehicles," in 2015 IEEE PELS Workshop on Emerging Technologies: Wireless Power (2015 WoW), pp. 1–6, 2015.

[19] T.-D. Yeo, D. Kwon, S.-T. Khang, and J.-W. Yu, "Design of maximum efficiency tracking control scheme for closed-loop wireless power charging system employing series resonant tank," IEEE Transactions on Power Electronics, vol. 32, no. 1, pp. 471–478, 2017.

[20] Z. Huang, S.-C. Wong, and C. K. Tse, "Control design for optimizing efficiency in inductive power transfer systems," IEEE Transactions on Power Electronics, vol. 33, no. 5, pp. 4523–4534, 2018.

[21] W. X. Zhong and S. Y. R. Hui, "Maximum energy efficiency tracking for wireless power transfer systems," IEEE Transactions on Power Electronics, vol. 30, no. 7, pp. 4025–4034, 2015.

[22] M. Fu, H. Yin, X. Zhu, and C. Ma, "Analysis and tracking of optimal load in wireless power transfer systems," IEEE Transactions on Power Electronics, vol. 30, no. 7, pp. 3952–3963, 2015.

[23] P. K. S. Jayathurathnage, A. Alphones, D. M. Vilathgamuwa, and A. Ong, "Optimum transmitter current distribution for dynamic wireless power transfer with segmented array," IEEE Transactions on Microwave Theory and Techniques, vol. 66, no. 1, pp. 346–356, 2018.

[24] S. M. Arif, T. T. Lie, B. C. Seet, S. Ayyadi, and K. Jensen, "Review of electric vehicle technologies, charging methods, standards and optimization techniques," Electronics, vol. 10, no. 16, 2021.

[25] S. A. Q. Mohammed and J.-W. Jung, "A comprehensive state-of-the- art review of wired/wireless charging technologies for battery electric vehicles: Classification/common topologies/future research issues," IEEE Access, vol. 9, pp. 19572–19585, 2021.

[26] S. A. Q. Mohammed and J.-W. Jung, "A comprehensive state-of-the- art review of wired/wireless charging technologies for battery electric vehicles: Classification/common topologies/future research issues," IEEE Access, vol. 9, pp. 19572–19585, 2021.

[27] M. Amjad, M. F. i Azam, Q. Ni, M. Dong, and E. A. Ansari, "Wireless charging systems for electric vehicles," Renewable and Sustainable Energy Reviews, vol. 167, p. 112730, 2022.

Date available online: 23-07-2025
Vol. 3, Issue 1 (January – June 2025)

DOI: https://doi.org/10.24312/ucp-jeit.03.01.515

Microservice Antipatterns: Causes, Detection, and Refactoring Challenges

Junaid Aziz[1], and Ghulam Rasool[1]
[1]Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan

Corresponding author: Junaid Aziz (e-mail: mjunaidaziz@gmail.com).

ABSTRACT
Microservice antipatterns negatively impact quality, necessitating a thorough understanding of their causes, effects, and solutions. This study provides a comprehensive review of antipatterns after analyzing 50 studies through a multivocal literature review. Key findings show that unprepared adoption and team culture are major causes, affecting maintainability, performance, and testing. Detection techniques are categorized into five groups, with most tools using search-based approaches. Four refactoring strategies were identified, along with their limitations. The study also highlights research gaps and challenges, guiding future work in improving detection and refactoring methods to mitigate antipattern effects.

INDEX TERMS: Microservice Architecture; Anti-patterns; Antipatterns Detection; Antipatterns Refactoring

I. INTRODUCTION

Despite the benefits of using Microservice architecture (MSA), there exist several open challenges, which can be grouped into two categories: technological and organizational [1]. Both are critical to the correct functioning of the system. However, these challenges may differ slightly when a large existing code base from monoliths is converted to microservices as compared to creating microservices from scratch. A study performed by Alshuqayran et al. [5] on MSA found communication, deployment operations, security, service discovery, and performance as major challenges of using microservices. Similarly, Jamshidi et al. [3] illustrated not only challenges faced by microservices but also envisioned the development of common microservice infrastructure through industry-academia collaboration to tackle such problems.

Recently, a discussion has emerged about taking a viewpoint of practitioners on the definition of antipatterns along with refactoring techniques and tools proposed in academic literature. Lacerda et al. [4] observed that the knowledge of developers about antipatterns detection and refactoring can not only help to improve tools but also the process of refactoring itself. Tahir et al. [5] also used data from the Stack Exchange website to identify the gap between what researchers and developers discuss about code smells and antipatterns? Based on their observations, they came to several conclusions: such as 1) most of the antipatterns detection tools only provide support for a few popular languages, 2) only developers can evaluate the level of antipatterns in a piece of code. Tian et al. [6] conducted an exploratory study to take the viewpoint of developers on architecture antipatterns by analyzing related discussions on Stack Overflow. Posts related to different architecture styles including microservices were extracted and analyzed. Results of their study indicate that detection and refactoring solutions must consider the causes of architectural antipatterns, and practitioners tend to use static code analysis tools to detect and refactor architectural antipatterns. Additionally, practitioners are concerned about the impact of architecture antipatterns on the performance and maintainability of the system and advocates for further research in this regard [7]. Moreover, they are also facing a lack of tool support in this regard. Considering these factors, it is important to mine the literature and identify all the causes and impact of antipatterns on microservices. This will help the community in building appropriate tools not only for dealing with factors causing antipatterns in microservices but also to address the concerns of practitioners about reducing level of different impacts of antipatterns i.e., performance on microservice-based applications. This multivocal literature review (MLR) tries to fill this gap by consolidating both academic and industrial knowledge. The objective of this MLR is to capture the state of art and practice on microservice antipatterns. The following are the major contributions of this study

- Provide an overview of types, causes, and impact of microservice antipatterns reported by both academia and industry
- Outline techniques and tools employed by both researchers and practitioners for the detection as well as correction of microservice antipatterns
- Bridge the gap between researchers and practitioners by revealing deficiencies in existing techniques and tools along with identifying potential research opportunities.

The remainder of this paper is arranged as follows. In Section 2 related work is discussed. Section 3 outlines the research methodology employed in this study. Section 4 presents the results along with a detailed discussion. Finally, Section 5 provides the conclusions drawn from the research.

II. RELATED WORK

There have been few attempts aiming at reviewing the state-of-the-art and current practices on microservice antipatterns. An overview of these studies is illustrated here along with a summary which is shown in Table 1.

Mumtaz et al. [9] performed a mapping study to

discuss different architecture antipatterns detection techniques and tools. They not only observed the lack of tools but also highlighted the need for the identification of software metrics and their thresholds for detecting microservice antipatterns. As per their recommendations, the applicability of these techniques and tools should be based on the software development industry's perspective. They emphasize doing empirical validations with real-world projects spanning many areas and programming languages in this regard.

Taibi et al. [10] identified several agreed microservice architectural patterns widely adopted and reported advantages along with disadvantages for each pattern. In their study, they pointed towards different emerging issues such as the impact of an increase in the number of microservices on the quality of the system, choice of most suitable DevOps tool, the existence of antipatterns, etc.

Ponce et al. [11] conducted MLR and grouped security antipatterns based on different properties such as confidentiality, integrity, and authenticity. They also presented a taxonomy of these antipatterns along with refactoring. However, a proposition of a tool capable of automatically detecting and refactoring security antipatterns in microservice-based applications is lacking in their work. Besides, empirical validation of their proposed refactoring solutions is also missing in their work.

Table 1: Summary of related systematic literature reviews

Study	Studies Reviewed	Study Focus	Search Period	Study Type
[9]	85	Detection of architecture antipatterns	1999-2019	SMS
[10]	42	Advantages and disadvantages of microservice patterns	2014-2017	SMS
[11]	58	Security antipatterns and refactoring	2014-2020	MLR
[12]	31	Visualizing Anti-Patterns in Microservices at Runtime	Not specified	SMS

Abbreviations: MLR, Multivocal literature review; SLR, Systematic literature review; SGLR, Systematic grey literature review; SMS, Systematic mapping study; TLR, Tertiary literature review.

In a mapping study performed by Parker et al. [12], it is analyzed how anti-patterns in microservices can be visualized from a dynamic perspective. Based on the findings, a gap between visualization and detection of microservice antipatterns is highlighted. It is also found that among all available tools proposed in academic literature, no single tool is completely capable of detecting and visualizing them. However, their study is lacking the analyses performed on tools contributed from the industry such as Jaeger [13] and Zipkin [14].

In this study we intend to identify causes as well as the impact of microservice antipatterns. This information will help researchers and practitioners in automating the process of detecting and refactoring microservice antipatterns as suggested by Tian et al. [6] and Aziz et al. [8]. Besides, information about techniques and tools currently applied for detection as well as refactoring of microservice antipatterns is synthesized. Additionally, the limitations of such techniques and tools are also highlighted.

III. RESEARCH METHODOLOGY

This Multivocal Literature Review (MLR) was conducted following the guidelines established by Garousi et al. [15], which are derived from the Systematic Literature Review (SLR) methodology proposed by Kitchenham et al. [16]. In accordance with these guidelines, the MLR process consists of three key stages: planning, conducting, and reporting the review. Figure 1 illustrates the steps involved in each stage. During planning stage, initially we set a goal for this study which is to capture the state of the art and practices in identifying types, causes, impact, detection, and refactoring techniques used for microservice antipatterns. Then, to make a decision about including grey literature in this study, a questionnaire is set (see Table 2) with possible answers either Yes, Maybe or No as per the guidelines by Garousi et al. [15]. Answer to each question is provided by authors through consensus. After consolidating the results, majority of questions are found to be responded in yes. This led us to the need of conducting a comprehensive MLR instead of a Systematic literature review to find answers to the following research questions:

- **RQ1:** What are the main causes that lead to antipatterns in microservices?
  Rationale - We want to explore the causes of antipatterns in microservices reported by researchers and practitioners.
- **RQ2:** How do antipatterns affect microservices?
  Rationale - We want to study the impact of antipatterns on microservice-based applications specifically on the process, performance, and people.
- **RQ3:** What techniques and tools are used for detecting antipatterns in microservices?
  Rationale - We want to learn about techniques and tools that are used by researchers and practitioners for the detection of antipatterns in microservices.
- **RQ4:** What are the refactoring techniques currently employed to resolve antipatterns in microservices?
  Rationale - We want to discover refactoring solutions proposed by researchers and practitioners to mitigate the effects of antipatterns in microservices.

During conducting stage, first, search string and data sources for academic and grey literatures were finalized as shown in Table 3. Then, authors conducted search for relevant academic and grey literature studies using respective data sources. The final selection of studies was made with consensus whereas conflicts were resolved through the mediation of another researcher. Following steps were performed for the search and selection of primary studies:

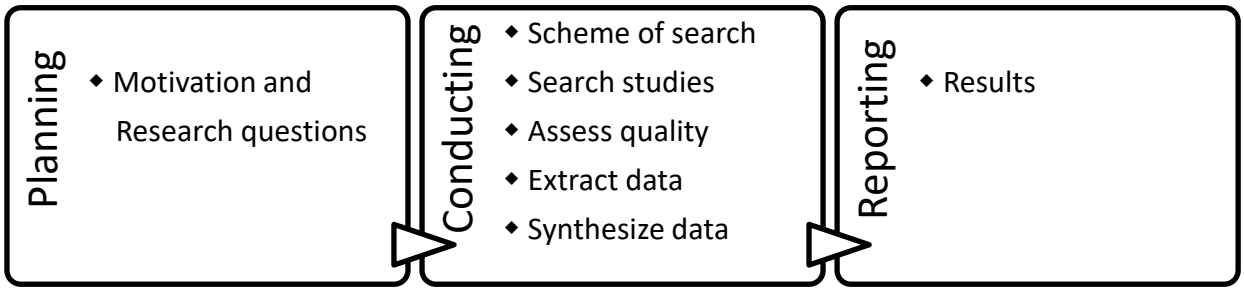

Figure 1: Steps followed for MLR

Table 2: Questionnaire used for including grey literature

Question	Response
Is academic literature not enough to provide solution for the research problem?	Yes
Is quality of evidence generated from academic literature lacking?	Maybe
Is finding context of the research problem in relation to practice necessary?	Maybe
Is this an attempt to validate research outcomes with experiences of practitioners or vice versa?	Yes
Is this an attempt to support research findings with practical experiences?	Yes
Is insights gained from academic and grey literature studies useful for one or both communities?	Yes
Is there an interest shown by practitioners in research problem through large number of contributions?	Yes

Step1 — Search process: For academic literature when we ran the search string, it provided 1032 results whereas for grey literature (see Table 3), it yielded 190,000 results on Google and 101,000 on Bing as shown on top of results page. DuckDuckGo was not providing this information on its results page. Initially, we limited our review of these results by title and abstract to the first 10 pages on every search engine. Afterwards, we gradually moved to the results on other pages until we found that at least half of the results on a page were not pertinent for this research. Duplicate results were also discarded at this stage.

Step2 — Quality assessment: The quality of the selected academic studies was assessed using the formula (1) opted by Ahmad et al. [17]. This formula is based on the recommendations presented in [18] for the qualitative assessment of selected studies. To calculate the quality score, the formula uses five general (i.e., QA1 to QA5) and five specialized assessment elements (i.e., QA6 to QA10) mentioned in Table 5. Since specific contributions of a study are more important than general factors for assessment, therefore, they are assigned 75% weight. An academic study was included if its accumulative quality score was greater than or equal to 1.5.

$$Quality\ Score = \left[\frac{\sum_{G=1}^{5}}{5} + \left(\frac{\sum_{S=1}^{5}}{5} \times 3\right)\right] \quad (1)$$

The quality of grey literature was assessed using the criteria suggested by Garousi et al. [15] as specified in Table 4. Every item of the criteria was assessed one by one for each study by the authors.

After consolidating the results, a grey literature study with a score of 8 or more (set through consensus) was included in the final list of primary studies with decent quality and rest were excluded. This provided us 33 academic (see Table A.2 in Appendix A) and 15 grey literature studies (see Table A.1 in Appendix A). After performing forward and backward snowballing on these studies, 2 further studies were found only for academic literature. We collaboratively extracted and encoded the necessary data from each of the selected primary studies using open and selective coding [69]. Initially, we extracted the metadata such as name, publication year, publication type (for academic literature) and contribution type (for grey literature). Table 6 provides a precise view of a complete list of defined metadata used in this study.

Table 3: Search string and data sources used in this MLR

Literature	Data Source	URL	Search String
Academic	IEEE Xplore	https://ieeexplore.ieee.org	(smell OR antipattern OR anti-pattern OR debt OR anomal OR refactor OR fault OR challeng OR vulnerab) AND (microservice OR micro-service)
	ACM Digital Library	https://dl.acm.org	
	Springer	https://link.springer.com	
	ScienceDirect	https://www.sciencedirect.com	
	DBLP	https://dblp.org	
	Scopus	https://www.scopus.com/	
Grey	Google	www.google.com	
	Bing	https://www.bing.com	
	DuckDuckGo	https://duckduckgo.com	

Table 4: Quality assessment for grey literature

Code	Items (Yes = 1 , No = 0)
QA1	Is this a reliable publisher or is the author belonged to a credible organization?
QA2	The author has published in the subject before
QA3	The author is an expert in the field
QA4	The claim of the sources is accurate
QA5	There are no ulterior motives
QA6	The data backs up the conclusions
QA7	The source has a defined goal
QA8	The source addresses a specific problem (s)
QA9	The methodology used by the source is explicitly explained
QA10	The source includes references to support the claims stated in the research
QA11	Limitations are clearly stated
QA12	The date of the source is clearly stated
QA13	The source talks about or links related GL or formal sources
QA14	The source enriches or adds something to the field of microservice antipatterns
QA15	The source supports or contradicts a current assertion
QA16	To support the claims presented in the study, the source includes citations and backlinks

Table 5: Quality assessment for academic literature

Code	Items (Yes = 1, Partial = 0.5, No = 0)
General	
QA1	Study provides problem definition and motivation
QA2	Research environment is clearly explained
QA3	Research methodology is presented in the study
QA4	Insights and lessons learned are explicitly mentioned
QA5	Contributions along with results are explicitly discussed
Specific	
QA6	Research focus is clearly on antipatterns in microservices
QA7	Study gives a clear picture of problems, solutions, and challenges concerning antipatterns in microservices
QA8	Research clearly states the validation technique applied on its outcome and relevant threats
QA9	Research presents techniques and tools used for the detection or correction of antipatterns in microservices
QA10	Study identifies new directions related to antipatterns in microservices

Table 6: Data Extraction Form: A = Academic Literature, G = Grey Literature

Search Criteria	Data Item	Description	Source
Demographic info	Study ID	Reference Code (AL, GL) sequentially incremented	A/G
	Name	Study title	A/G
	Publication Type	J=Journal, C=Conference	A
	Study Aim	Personal notes about each study	A/G
	Year	Publication year	A/G
	URL	URL of publication	A/G
	Contribution Type	Blog post, Industrial whitepaper, Video, Article, Book	G
RQ1	Causes	Causes of antipatterns	A/G
	Study Design	Experimental, Empirical Study, Solution proposal, Case Study, Personal experience, Tool	A/G
RQ2	Impact	Impact of antipatterns	A/G
RQ3	Detection of antipatterns	Techniques, tools applied for detection of antipatterns in microservices	A/G
RQ4	Refactoring	Solutions to resolve antipatterns in microservices	A/G

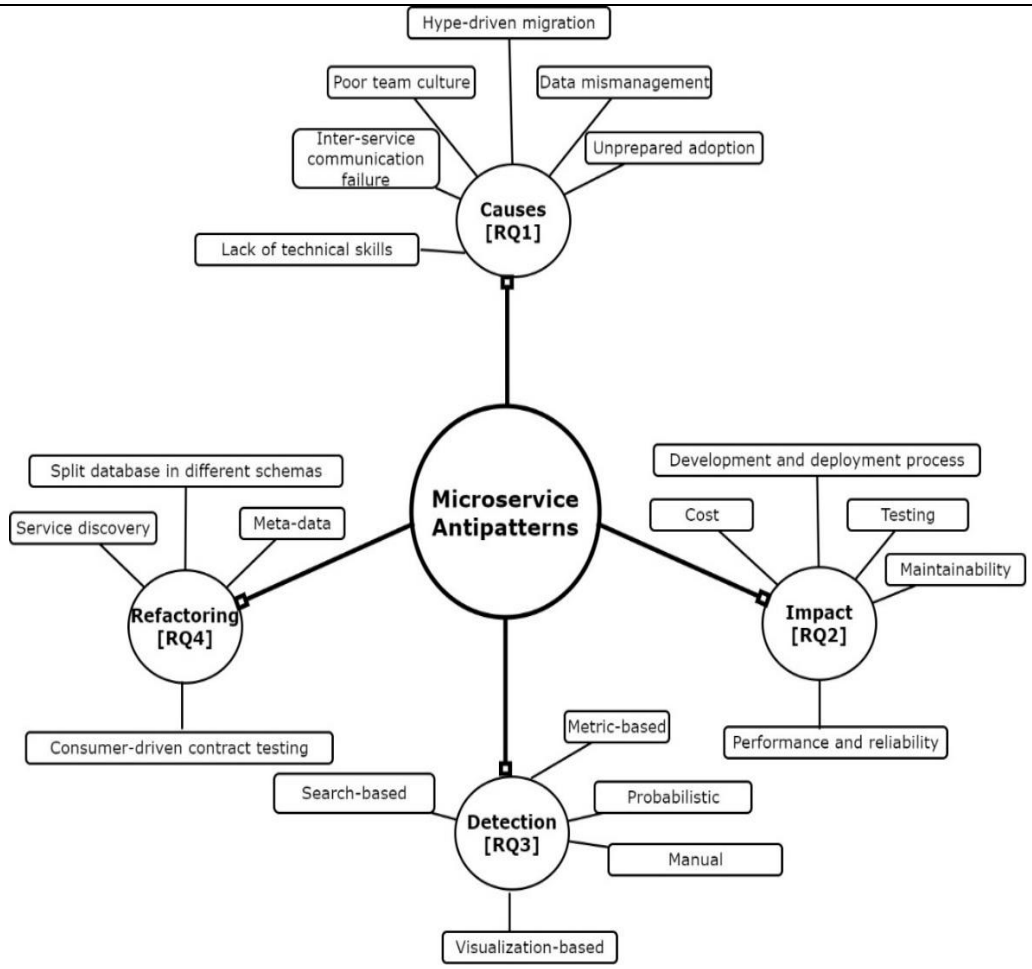

Figure 2: Overview of the study

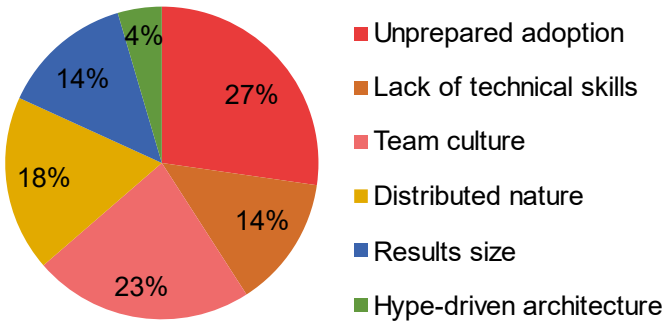

Figure 3: Causes of microservice antipatterns

IV. RESULTS AND DISCUSSION

This section presents and discusses the key findings derived from evaluating and synthesizing the selected studies in relation to each research question. Figure 2 summarizes the key findings of our RQs.

A. RQ1: What are the main causes that lead to antipatterns in microservices?

This study has identified various reasons that introduce antipatterns in microservices. These reasons are extracted from the studies based on industrial surveys and experiences gained from migration of legacy systems to MSA. We classified them in a list below and also summarized them in Figure 3. From the list, some of the causes are found to be overlapping with the causes of antipatterns in other type of applications as mentioned in [70]. However, distributed nature and hype-driven architecture are found applicable only to MSA.

- **Unprepared adoption:** Before deciding to migrate to MSA, organizations need to think about the extra effort required to work on automated deployment, monitoring, failure, eventual consistency, and other issues that this architecture style introduces. Many authors (A1, A5, A7, A21, G3,G13) have mentioned this cause in their studies.
- **Lack of technical skills:** Microservices adoption is

difficult since it demands knowledge of new techniques and technologies, as well as the requirement to automate software deployment and monitoring operations (A7, A9, A22). Therefore, the development team with poor technical skills and their lack of awareness of the tools involved can easily cause antipatterns in such types of applications as highlighted in (A32).

- **Team culture:** Many authors (A5, A7, A20, G3, G13) have cited practices and culture that exist in teams or organizations as a cause of antipatterns in microservices. For instance, teams must be organized to handle microservices independently. Otherwise, this may lead to the development and deployment-related issues.
- **Distributed nature:** The current literature (A3,A18,G4,G15) indicates that MSA implies some challenging problems of data integrity, management of microservices, and their consistency due to its distributed nature. These challenges can lead to different types of antipatterns in applications.
- **Results size:** In (A18, A20, G4), the authors highlighted the need for an efficient and consistent database management solution, especially in data-driven microservices applications that process large amounts of data. According to them, this can seriously affect the performance of the software.
- **Hype-driven architecture:** Shifting to MSA because of its popularity only, believing that all your software-related problems will be solved may cause antipatterns and affect the quality of the software (A22).

The implementation of microservices require that organizations have empowered small teams handling them in a way that corresponds to particular business domains (A5). Product owners, architects, developers, quality engineers, and operational engineers are just a few of the roles that these teams must include in order to provide these services. Complicated code bases are typically produced by large teams, which hinders and delays future updates (A21, G3). In addition, teams that lack authority frequently face delays while they wait for higher-ups to make decisions. Similarly, a lack of alignment with business goals results in dependencies between teams, leading to more delays and drifting away from MSA. It is crucial for teams to possess essential skills like API design, development, and understanding of distributed applications (A7). Without these skills, there might be increased costs for training or hiring, potentially diverting the solution from the microservices approach as teams fall back on their familiar technologies (A9).

The culture within an organization significantly influences its approach to working on systems, as it shapes the individual decisions made by its members. In project-centric cultures, teams are assembled to tackle specific issues, disbanding once the problem is resolved. However, this practice often results in a loss of valuable knowledge about both the problem and its solution. Additionally, if there is a need to revisit the same problem or make further modifications to a component, reconstructing the team or recovering lost knowledge can pose challenges. In the case of microservices, it becomes quite difficult for an organization if it decides to operate in project-centric culture instead of offering team-based ownership of components (A20, G3). Similarly, whenever outsourcing is conducted; the outsourced team can neither adopt the desired organizational culture as it is nor it can avoid such change. This suggests that culture plays a crucial role in determining the suitability of companies or individuals selected to endorse the outsourcing model.

Implementation of MSA brings a lot of challenges and it doesn't work for every organization. Some organizations may find it the only way to keep up with rapid development and deliver software on time. In general, organizations that mismatch any of these causes which are discussed here will pay a toll in the form of antipatterns when attempting to apply MSA. So, instead of just following the hype, the decision about such a transition should be made after evaluating its cost and gains (A22).

B. RQ2: How do antipatterns affect microservices?

Antipatterns in microservices can harm the resulting applications if suitable techniques and processes are not followed. Different authors have mentioned those impacts which are shown in Figure 4.

Microservices need to be as autonomous and decoupled as possible to ease the development and deployment process (G14). This implies that using norms and standards for the definition of microservices contracts can make a huge impact on resulting applications (A13, A19).

Centering microservices around technical concerns only can easily mutate into something called layering which was the main disadvantage of service-oriented architecture. This can eventually incur performance and reliability issues such as high network latency, failures of one service affecting others, and slowing down the whole development process (A1, A3, G6). In (A31), authors performed an experimental study to validate the existence of correlation between microservice antipatterns and microservices performance. Based on their findings, Cyclic Dependency and Shared Persistence have significant negative effect on the performance of microservices. More such studies are needed to find out impact of different microservice antipatterns on metrics such as performance, maintenance, cost etc.

Developers frequently construct test cases (e.g., unit, integration, and system test cases) without having a complete understanding of the operational environment or user behavior. Faults are typically identified only during the process of updating from one service version to another, or occasionally after the service update has been completed (A23, A46, G1).

Implementation of MSA requires organizations to adopt DevOps. Lack of such practice can cause different forms of antipatterns such as API Versioning, Human Evolvability, and Lack of evaluation methods. These antipatterns will eventually hinder the maintainability of the system (A19, A24, A32, G6). Furthermore, numerous instances of a service might be active at the same time. Using virtualization to deploy them isn't cost-effective,

and it also adds a lot of processing overhead (A19).

The microservices approach offers significant benefits, but can be expensive as it requires different infrastructure. Also, additional code is often required to communicate between services in the form of API calls. Bad antipatterns such as Cyclic Dependency that are introduced in such calls can make a great impact on the development and deployment of these services (A13). Besides, systems such as service directories, messaging, and queuing services are required to identify appropriate services and then route calls to them. The presence of antipatterns in any of these can affect performance and reliability (A1,A3,A34,A35). Since microservices can be scripts, containers, or entire virtual machines and require a structured way to package and deploy them, the introduction of antipatterns like Red Flag can make a huge impact on them (A24, G6).

Applications based on MSA also need to implement a system for monitoring service performance and behavior, along with specific error handling techniques. If a microservice is not responding then there is no easy way for other services to understand the error or determine the problem. Additional code and monitoring are required to ensure that problems are treated as errors rather than simply piling them up (A23). A mechanism to decide when to include features in services and when to split features into separate services is also required. Otherwise, these services will be affected by antipatterns like Wrong Cut (A46, G1). Moreover, it has been found that the impact of antipatterns on the cost of developing microservices has been studied in (A19) only. Further research in this area especially providing cost comparison of deployment of the microservices-based application on different cloud containers will be effective and helpful.

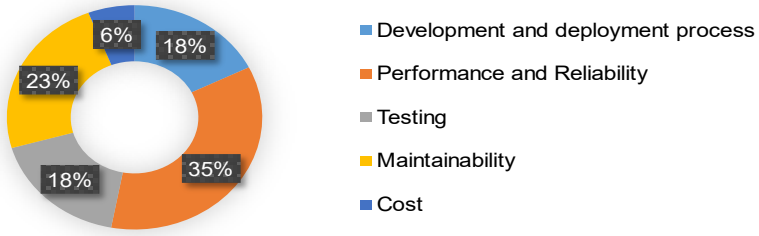

Figure 4: Impact of microservice antipatterns

C. RQ3: What techniques and tools are used for detecting antipatterns in microservices?

Based on the literature review, we have identified techniques used for the detection of antipatterns in microservices. These techniques are further classified into five broad categories (i.e., manual approaches, metric-based approaches, probabilistic approaches, visualization-based approaches, and search-based approaches) as proposed by Kessentini et al. [71]. The list of antipatterns detected by respective category along with the information about tools used for detection is shown in Table 7.

- **D1: Search-based approaches:** Source code and/or bytecode analysis are used to create a realistic representation of the application. This includes a tree representation, identification of the system's endpoints, and the creation of a communication map. Antipatterns in microservices can be detected with the help of these representations. With the use of performance monitoring data, some of the techniques in this category use various machine learning algorithms to classify the actions of the target system. In the first phase, fault injection is used to collect samples of labeled performance data reflecting various service behavior, and multiple classification models are trained using this data. In the second phase, real-time performance data is transferred to these models for the accurate detection of anomalies.
- **D2: Metric-based approaches:** Typically, this type of approach revolves around building an architectural model by employing the reverse engineering approach on source code and communications logs. This model is then used to evaluate the architecture design based on identified principles with corresponding metrics. A dependency graph or other methods can be used to visualize the results.
- **D3: Visualization-based approaches:** This type of approach for detecting antipatterns is about collecting all service invocation links and constructing a service dependency graph representing them. A visual representation of the graph is generated that allows users to browse all service dependency relationships and check for anomalies or errors.
- **D4: Manual approaches:** The information stored in a microservice catalog helps teams not only resolve their production incidents quickly but also build reliable and more operable microservices. It helps to track all the services and systems running in production. Without changing the user code, data regarding resource usage, performance counters, power consumption, and network performance is collected. The information is then utilized to evaluate microservice-based systems in terms of their interactions with the outside world as well as internal connections and dependencies.
- **D5: Probabilistic approaches:** Service workload is continuously monitored and its response time is regularly compared with baseline response time. After applying standard statistical outlier detection techniques, a higher deviation between these two means that the service is suspected to have performance anomalies.

In academic literature, search-based approaches using static code analysis were found to be effective as compared to other techniques because these helped researchers detect a large number of antipatterns in microservices. Besides, tools used for detection have also been made available by them. Moreover, metrics-

based and visualization-based approaches have been experimented on the detection of diverse antipatterns unlike others (i.e., manual and probabilistic approaches) which have been applied for the detection of monitoring type of antipatterns only. In grey literature, microservice catalog is the only technique from the category of manual approaches found to be effective for the detection of a limited number of antipatterns in microservices. Big companies such as LinkedIn, Spotify, Shopify, Bell, etc. have relied on in-house built catalogs but have not made them available online.

D. RQ4: What are the refactoring techniques currently employed to resolve antipatterns in microservices?

Refactoring is a technique that is applied to reorganize the structure of the application without altering its original behavior. It is often performed to remove antipatterns and improve design quality. Even though refactoring is now a common practice in the industry, manual refactoring of antipatterns is still a risky and error-prone task, especially when it is performed by inexperienced people in a team.

Table 7: Antipatterns detection techniques and tools

Technique	Reference Study	Tool Name	Release Type	Languages	Accuracy (%)	Antipatterns Detected	Validation Set
D1	A9	-	-	-	-	LEM	-
	A14	TraceAnomaly	Open Source	Java	P=98, R=97	TA	792 Ms
	A16	-	-	-	-	TA	-
	A25	MEPFL	Experimental	Java	Average P=89 Average R=82	TA	49 Ms
	A26	ARCAN	Prototype	Java	P=100	CD,SP,HE	40 Ms
	A30	MSANose	Open Source	Java	Not measured	WC,NG,SG,CD,SL,LTS, SV,SP, ISI,EU,HE	45 Ms
	A33	MARS	Open Source	Java	Average P=82 Average R=89	WC,NG,CD,SL,SV,SP,HCE	171 Ms
D2	A4	-	-	-	-	SG,CD	-
	A15	-	-	-	-	TA	-
	A27	-	-	-	-	LM	-
D3	A2	-	-	-	-	NST,CD	-
	A6	-	-	-	-	CD,DM,IS	-
D4	G10	-	-	-	-	LTS,LG,HE	-
	G12	-	-	-	-	CD,SP	-
	A34	DEEP-mon	Open Source	Golang,C++	Not measured	LM	36 Ms
D5	A17	-	Experimental	-	Not measured	TA	-

Abbreviations: SP, Shared Persistence; HE, Hard-coded Endpoints; DM, Distributed Monolith; TA, Trace anomaly; CD, Cyclic Dependency; LEM, Lack of evaluation methods; WC, Wrong Cuts; NG, No-API Gateway; SG, Microservice Greedy; SL, Shared Libraries; LTS, Too Many Standards; SV, API Versioning; ISI, Inappropriate Service Intimacy; EU, ESB Usage; LM, Lack of monitoring; NST, No Service Template; IS, Influential Service; LG, Lack of guidance; HE, Human Evolvability; Ms, Microservices; P, precision; R, recall.

Moreover, assigning resources to perform refactoring for microservice antipatterns is not always feasible, due to constraints in the budget, shorter release cycles, and staff shortage. Therefore, researchers and practitioners use different strategies to automate the refactoring process. After reviewing the primary studies, the authors identified the following refactoring strategies:

- **R1: Split database in different schemas:** In (A28, G2, G5) authors have found this refactoring useful for resolving antipatterns like Shared Persistence and Distributed Monolith. This approach can be applied to different situations especially when services access the same data store. For instance, in a scenario where a portion of the data store is accessed by just one service, one way out is to split it into two different data stores, with one storing the portion of data accessed by that single service and the other storing the rest of the data.
- **R2: Consumer-driven contract testing:** In this technique, services communicate with one another using contracts that the consumer creates and then shares with the provider for verification. In most cases, the contract defines a series of transactions between the consumer and the provider. This type of testing technique has proven to be effective in evaluating service integrations. Consumer-driven contract testing, unlike end-to-end testing, can catch all types of errors because it is always done in isolation from other services (A10, G8, G9, G11). The occurrence of antipatterns like Oracle Problem and Test Endpoints can be avoided with the adoption of this strategy
- **R3: Meta-data:** Many authors (A1, A12, A23, A29) have made use of this approach in the form of data flow diagrams, data structures, tests derived from service operation data, microservice usage data,

etc. to resolve antipatterns like Wrong Cuts, Test Endpoints, and Oracle Problem.

• **R4: Service Discovery:** Antipatterns like Hard-coded Endpoints can be resolved with the help of this refactoring which normally occurs in an application when a service A directly invokes another service B either because the location of B is hardcoded in the source code of A, or no message router is used (A28, G7). It may also be possible to dynamically resolve the endpoint of service by simply adding a service discovery mechanism.

Table 8 summarizes the list of techniques that have been applied by researchers and practitioners for refactoring appropriate antipatterns in microservices. Only studies that have provided information about tools either implemented or used are shown in this table. Despite these efforts, it is found that refactoring microservice antipatterns is still at an infancy level. Authors have also witnessed that more interest from the community is shown toward refactoring test antipatterns. Newman [3] also highlight the challenges of performing end-to-end testing on microservices.

In (G11), the author analyzes different types of testing used for microservices and based on personal experience suggests contract testing as a suitable choice. Besides, based on lessons learned from a case study (A10), the authors suggest that consumer-driven contract testing is a feasible practice, especially when dealing with microservices-based applications. Other approaches proposed in (A23, A29) which make use of run-time data of microservices may also help refactor test antipatterns. Refactoring approaches to resolve antipatterns violating key design principles of microservices such as horizontal scalability, isolation of failures, and decentralization is presented in (A28). A prototype is also implemented based on the methodology proposed with limited experimentation.

Table 8: Antipatterns refactoring techniques and tools

Reference	Refactoring Technique				Antipatterns Resolved					Tool Name	Release Type
	R 1	R2	R3	R4	SP	HE	DM	OP	TE		
A28	✓			✓	✓	✓	✓			µFreshener	Prototype
G2	✓						✓			Gremlin	Commercial
G8		✓						✓	✓	Postman	Commercial
A11		✓						✓	✓	Pactflow	Open Source /Commercial
A23			✓					✓	✓	ExVivoMicroTest	Prototype

Abbreviations: SP, Shared Persistence; HE, Hard-coded Endpoints; DM, Distributed Monolith; OP, Oracle Problem; TE, Test Endpoints.

E. Emerging Challenges and Research Opportunities

The study of microservice antipatterns remains an emerging and rapidly evolving research area. Through this investigation, we have identified following key challenges that present opportunities for future research and practical implementation:

• **Many antipatterns still need detection**

Our investigation uncovers that microservice antipatterns manifest not just during development phases, but can also become institutionalized at the organizational level in the absence of effective policy frameworks. The current list of microservice antipatterns detection tools have been developed with a focus on a limited number of antipatterns. Exposure to diverse antipatterns for such tools is needed. Moreover, current and new tools are required to be evaluated on medium to large scale industrial-based microservice systems as it is revealed in this study that the presence of bad antipatterns can impact on performance, maintainability, and testability of microservices.

• **Need to apply other techniques for detection of antipatterns**

This study finds that currently available antipatterns detection tools have mostly made use of search-based techniques only. Researchers need to explore other techniques for antipatterns detection and build corresponding tools. In this regard, metric-based and visualization-based techniques can be experimented with in detecting diverse types of antipatterns. This will also provide an opportunity for finding a more appropriate one, once the results of applying different techniques become available.

• **More research on identifying impact of bad antipatterns on microservice quality needed**

Initial studies suggest a potential relationship between the presence of certain microservice antipatterns and the overall quality of a microservices-based system (A31). However, to establish a comprehensive understanding, further research in this direction is needed. Investigating and quantifying this correlation can provide valuable insights for researchers and practitioners aiming to enhance the design and maintainability of microservices-based systems.

V. CONCLUSION

This study offers a concise yet thorough review of microservice antipatterns by analyzing academic and industry literature between 2014 and 2023. Key findings include:

• Identifying 6 root causes behind antipattern occurrences.
• Assessing their impacts on microservices.
• Classifying detection methods into 5 groups (manual, metric-based, probabilistic, visualization-based, and search-based).
• Highlighting research gaps and opportunities.
• Discovering 4 refactoring approaches for mitigation.

Current research primarily focuses on architecture, design, and organizational antipatterns, with limited tools available. Besides most tools detect only specific type of microservice antipatterns. Additionally, proposed refactoring methods often lack real-world testing.

A comprehensive, multi-language detection tool remains an unmet need in the field.

Appendix A. Studies selected for this MLR

Table A.1: Selected studies in grey literature

ID	Quality Assessment Codes (QA)																Quality score	Reference
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
G1	1	0	0	0	1	0	1	0	1	0	0	1	0	1	1	1	8	[49]
G2	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	9	[50]
G3	1	1	0	0	1	0	1	0	0	1	1	1	1	1	1	1	11	[51]
G4	0	1	0	0	0	0	1	0	1	1	1	1	1	1	1	0	9	[52]
G5	1	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	10	[53]
G6	0	1	0	1	0	1	1	0	0	1	1	1	0	1	1	1	10	[54]
G7	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0	1	11	[55]
G8	0	1	0	1	1	0	0	1	0	1	0	1	1	1	1	0	9	[56]
G9	1	0	0	1	1	0	1	0	0	1	1	1	1	1	1	1	11	[57]
G10	1	0	0	1	0	1	1	1	0	1	1	1	1	1	1	1	12	[58]
G11	0	1	0	0	1	0	1	0	1	1	0	1	1	1	0	1	9	[59]
G12	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	8	[60]
G13	1	0	0	1	0	1	1	1	1	1	1	1	1	1	0	0	11	[63]
G14	1	0	0	1	1	1	1	0	1	0	0	1	1	0	0	1	9	[64]
G15	0	1	1	0	0	1	1	0	1	1	1	0	0	1	0	0	8	[68]

Table A.2: Selected studies in academic literature

ID	Title	Year	Quality score	Reference
A1	Challenges of Production Microservices	2018	2.6	[19]
A2	Graph-based and scenario-driven microservice analysis, retrieval, and testing	2019	2.9	[20]
A3	Microservice Disaster Crash Recovery: A Weak Global Referential Integrity Management	2020	3.2	[21]
A4	Evaluation of Microservice Architectures: A Metric and Tool-Based Approach	2018	2.6	[22]
A5	Exploring the Microservice Development Process in Small and Medium-Sized Organizations	2020	2.9	[23]
A6	Service Dependency Graph Analysis in Microservice Architecture	2020	3.4	[24]
A7	An Experience Report from the Migration of Legacy Software Systems to Microservice Based Architecture	2019	2.7	[25]
A8	Tool Support for the Migration to Microservice Architecture: An Industrial Case Study	2019	2.6	[26]
A9	Anomaly Detection and Diagnosis for Container-Based Microservices with Performance Monitoring	2018	3.5	[27]
A10	Consumer-Driven Contract Tests for Microservices: A Case Study	2019	3.2	[28]
A11	Fine-Grained Access Control for Microservices	2018	3.3	[29]
A12	From Monolith to Microservices: A Dataflow-Driven Approach	2017	3.5	[30]
A13	Functional-First Recommendations for Beneficial Microservices Migration and Integration Lessons Learned from an Industrial Experience	2019	3.3	[31]
A14	Unsupervised Detection of Microservice Trace Anomalies through Service-Level Deep Bayesian Networks	2020	3	[32]
A15	Self-Adaptive Root Cause Diagnosis for Large-Scale Microservice Architecture	2020	3.5	[33]
A16	An Intelligent Anomaly Detection Scheme for Micro-Services Architectures With Temporal and Spatial Data Analysis	2020	2.6	[34]
A17	RAD: Detecting Performance Anomalies in Cloud-based Web Services	2020	3	[35]
A18	Framework for Interaction Between Databases and Microservice Architecture	2019	2.1	[36]
A19	Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture	2016	3.6	[37]
A20	An Expert Interview Study on Areas of Microservice Design	2018	3.3	[38]
A21	Migrating Towards Microservice Architectures: An Industrial Survey	2018	2.7	[39]
A22	An Experience Report on the Adoption of Microservices in Three Brazilian Government Institutions	2018	3	[40]
A23	Automatic Ex-Vivo Regression Testing of Microservices	2020	3.1	[41]
A24	Integrating Continuous Security Assessments in Microservices	2017	2.9	[42]

	and Cloud Native Applications			
A25	Latent Error Prediction and Fault Localization for Microservice Applications by Learning from System Trace Logs	2019	2.6	[43]
A26	Towards Microservice Antipatterns Detection	2020	2.5	[44]
A27	Towards a method for monitoring the coupling evolution of microservice-based architectures	2020	3.3	[45]
A28	Freshening the Air in Microservices: Resolving Architectural Antipatterns via Refactoring	2020	2.9	[46]
A29	Testing microservice architectures for operational reliability	2020	3.6	[47]

Table A.2 (continued)

ID	Title	Year	Quality score	Reference
A30	Automated Code-Smell Detection in Microservices Through Static Analysis: A Case Study	2020	3.1	[48]
A31	An Empirical Study on Underlying Correlations between Runtime Performance Deficiencies and “Bad Antipatterns” of Microservice Systems	2021	2.5	[61]
A32	Impacts, causes, and solutions of architectural antipatterns in microservices: An industrial investigation	2022	3.7	[62]
A33	On the maintenance support for microservice-based systems through the specification and the detection of microservice antipatterns	2023	3.9	[65]
A34	Identifying Anti-Patterns in Distributed Systems With Heterogeneous Dependencies	2023	2.6	[66]
A35	An Approach for Evaluating the Potential Impact of Anti-Patterns on Microservices Performance	2023	2.8	[67]

References

[1] Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems (1st ed.). O’Reilly Media.

[2] Alshuqayran, N., Ali, N., & Evans, R. (2016, November). A systematic mapping study in microservice architecture. In 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA) (pp. 44-51). IEEE.

[3] Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The journey so far and challenges ahead. IEEE Software, 35(3), 24-35.

[4] Lacerda, G., Petrillo, F., Pimenta, M., & Guéhéneuc, Y. G. (2020). Code antipatterns and refactoring: A tertiary systematic review of challenges and observations. Journal of Systems and Software, 167, 110610.

[5] Tahir, A., Dietrich, J., Counsell, S., Licorish, S., & Yamashita, A. (2020). A large scale study on how developers discuss code antipatterns and anti-pattern in Stack Exchange sites. Information and Software Technology, 125, 106333.

[6] Tian, F., Liang, P., & Babar, M. A. (2019). How Developers Discuss Architecture Antipatterns? An Exploratory Study on Stack Overflow. 2019 IEEE International Conference on Software Architecture (ICSA).

[7] Zhou, X., Li, S., Cao, L., Zhang, H., Jia, Z., Zhong, C., ... & Babar, M. A. (2023). Revisiting the practices and pains of microservice architecture in reality: An industrial inquiry. Journal of Systems and Software, 195, 111521.

[8] Aziz, J., & Rasool, G. (2024). A Design-Oriented Classification of Microservice Smells. *UCP Journal of Engineering & Information Technology (HEC Recognized-Y Category)*, *2*(2), 33-40.

[9] Mumtaz, H., Singh, P., & Blincoe, K. (2020). A systematic mapping study on architectural antipatterns detection. *Journal of Systems and Software*, 110885.

[10] Taibi, D., Lenarduzzi, V., & Pahl, C. (2019). Continuous Architecting with Microservices and DevOps: A Systematic Mapping Study. Cloud Computing and Services Science, 126–151.

[11] Ponce, F., Soldani, J., Astudillo, H., & Brogi, A. (2022). Antipatterns and refactorings for microservices security: A multivocal literature review. Journal of Systems and Software, 192, 111393. doi:10.1016/j.jss.2022.111393

[12] Parker, G., Kim, S., Al Maruf, A., Cerny, T., Frajtak, K., Tisnovsky, P., & Taibi, D. (2023). Visualizing Anti-Patterns in Microservices at Runtime: A Systematic Mapping Study. IEEE Access.

[13] Jaeger. (n.d.). Retrieved April 30, 2023, from https://www.jaegertracing.io/

[14] Zipkin. (n.d.). Retrieved April 30, 2023, from https://zipkin.io/

[15] Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, *106*, 101-121.

[16] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software engineering," in "EBSE Technical Report," 2007, vol. EBSE- 2007-01

[17] Ahmad, A., Babar, M.A., 2016. Software architectures for robotic systems: A systematic mapping study. J. Syst. Softw. 122 (12), 16–39.

[18] Brereton, P., Kitchenham, B., Budgen, D., Turner, M., Khalil, M., 2007. Lessons from applying the systematic literature review process within the software engineering Domain. J. Syst. Softw. 80 (4), 571–583.

[19] Götz, B., Schel, D., Bauer, D., Henkel, C., Einberger, P., & Bauernhansl, T. (2018). Challenges of production microservices. *Procedia CIRP*, *67*, 167-172.

[20] Ma, S. P., Fan, C. Y., Chuang, Y., Liu, I. H., & Lan, C. W. (2019). Graph-based and scenario-driven microservice analysis, retrieval, and testing. *Future Generation Computer Systems*, *100*, 724-735.

[21] Manouvrier, M., Pautasso, C., & Rukoz, M. (2020, June). Microservice Disaster Crash Recovery: A Weak Global Referential Integrity Management. In *International Conference on Computational Science* (pp. 482-495). Springer, Cham.

[22] Engel, T., Langermeier, M., Bauer, B., & Hofmann, A. (2018, June). Evaluation of microservice architectures: A metric and tool-based approach. In *International Conference on Advanced Information Systems Engineering* (pp. 74-89). Springer, Cham.

[23] Sorgalla, J., Sachweh, S., & Zündorf, A. (2020, November). Exploring the microservice development process in small and medium-sized organizations. In *International Conference on Product-Focused Software Process Improvement* (pp. 453-460). Springer, Cham.

[24] Gaidels, E., & Kirikova, M. (2020, September). Service Dependency Graph Analysis in Microservice Architecture. In *International Conference on Business Informatics Research* (pp. 128-139). Springer, Cham.

[25] da Silva, H. H. S., Carneiro, G. D. F., & Monteiro, M. P. (2019). An experience report from the migration of legacy software

systems to microservice based architecture. In *16th International Conference on Information Technology-New Generations (ITNG 2019)* (pp. 183-189). Springer, Cham.

[26] Pigazzini, I., Fontana, F. A., & Maggioni, A. (2019, September). Tool support for the migration to microservice architecture: An industrial case study. In *European Conference on Software Architecture* (pp. 247-263). Springer, Cham.

[27] Du, Q., Xie, T., & He, Y. (2018). Anomaly Detection and Diagnosis for Container-Based Microservices with Performance Monitoring. Algorithms and Architectures for Parallel Processing, 560–572.

[28] Lehvä, J., Mäkitalo, N., & Mikkonen, T. (2019, November). Consumer-driven contract tests for microservices: A case study. In *International Conference on Product-Focused Software Process Improvement* (pp. 497-512). Springer, Cham.

[29] Antonio, N., Vitor, J., Khaled, M., & Ali, A. (2018, October). Fine-Grained Access Control for Microservices. In The 11th International Symposium on Foundations & Practice of Security (Vol. 11358). Springer.

[30] Chen, R., Li, S., & Li, Z. (2017, December). From monolith to microservices: A dataflow-driven approach. In *2017 24th Asia-Pacific Software Engineering Conference (APSEC)* (pp. 466-475). IEEE.

[31]Gouigoux, J. P., & Tamzalit, D. (2019, March). "Functional-First" Recommendations for Beneficial Microservices Migration and Integration Lessons Learned from an Industrial Experience. In *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)* (pp. 182-186). IEEE.

[32] Liu, P., Xu, H., Ouyang, Q., Jiao, R., Chen, Z., Zhang, S., Yang, J., Mo, L., Zeng, J., Xue, W., & Pei, D. (2020). Unsupervised Detection of Microservice Trace Anomalies through Service-Level Deep Bayesian Networks. 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE).

[33] Ma, M., Lin, W., Pan, D., & Wang, P. (2020). Self-Adaptive Root Cause Diagnosis for Large-Scale Microservice Architecture. *IEEE Transactions on Services Computing*.

[34] Zuo, Y., Wu, Y., Min, G., Huang, C., & Pei, K. (2020). An intelligent anomaly detection scheme for micro-services architectures with temporal and spatial data analysis. *IEEE Transactions on Cognitive Communications and Networking*, *6*(2), 548-561.

[35] Mukherjee, J., Baluta, A., Litoiu, M., & Krishnamurthy, D. (2020, October). RAD: Detecting Performance Anomalies in Cloud-based Web Services. In *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)* (pp. 493-501). IEEE.

[36] El Kholy, M., & El Fatatry, A. (2019). Framework for interaction between databases and microservice architecture. *IT Professional*, *21*(5), 57-63.

[37] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables devops: Migration to a cloud-native architecture. *Ieee Software*, *33*(3), 42-52.

[38] Haselböck, S., Weinreich, R., & Buchgeher, G. (2018, November). An expert interview study on areas of microservice design. In *2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA)* (pp. 137-144). IEEE.

[39] Di Francesco, P., Lago, P., & Malavolta, I. (2018, April). Migrating towards microservice architectures: an industrial survey. In *2018 IEEE International Conference on Software Architecture (ICSA)* (pp. 29-2909). IEEE.

[40] Luz, W., Agilar, E., de Oliveira, M. C., de Melo, C. E. R., Pinto, G., & Bonifácio, R. (2018, September). An experience report on the adoption of microservices in three Brazilian government institutions. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (pp. 32-41).

[41] Gazzola, L., Goldstein, M., Mariani, L., Segall, I., & Ussi, L. (2020, October). Automatic ex-vivo regression testing of microservices. In *Proceedings of the IEEE/ACM 1st International Conference on Automation of Software Test* (pp. 11-20).

[42] Torkura, K. A., Sukmana, M. I., & Meinel, C. (2017, December). Integrating continuous security assessments in microservices and cloud native applications. In *Proceedings of the10th International Conference on Utility and Cloud Computing* (pp. 171-180).

[43] Zhou, X., Peng, X., Xie, T., Sun, J., Ji, C., Liu, D., ... & He, C. (2019, August). Latent error prediction and fault localization for microservice applications by learning from system trace logs. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 683-694).

[44] Pigazzini, I., Fontana, F. A., Lenarduzzi, V., & Taibi, D. (2020, June). Towards microservice antipatterns detection. In *Proceedings of the 3rd International Conference on Technical Debt* (pp. 92-97).

[45] de Freitas Apolinário, D. R., & de França, B. B. N. (2020, October). Towards a method for monitoring the coupling evolution of microservice-based architectures. In *Proceedings of the 14th Brazilian Symposium on Software Components, Architectures, and Reuse* (pp. 71-80).

[46] Brogi A., Neri D., Soldani J. (2020) Freshening the Air in Microservices: Resolving Architectural Antipatterns via Refactoring. In: Yangui S. et al. (eds) Service-Oriented Computing – ICSOC 2019 Workshops. ICSOC 2019. Lecture Notes in Computer Science, vol 12019. Springer, Cham.

[47] Pietrantuono, R., Russo, S., & Guerriero, A. (2020). Testing microservice architectures for operational reliability. *Software Testing, Verification and Reliability*, *30*(2), e1725.

[48] Walker, A., Das, D., & Cerny, T. (2020). Automated code-smell detection in microservices through static analysis: A case study. *Applied Sciences*, *10*(21), 7800.

[49] Mahran, L. (2020). TESTING MICROSERVICES: PRINCIPLES, CHALLENGES, CASE STUDIES [Blog]. Retrieved 2024, from https://mobidev.biz/blog/testing-microservices-principles-challenges-case-studies.

[50] Newman, A. (2020). Is your microservice a distributed monolith? [Blog]. Retrieved 2021, from https://www.gremlin.com/blog/is-your-microservice-a-distributed-monolith/.

[51] Grabner, A. (2016). *Locating Common Micro Service Performance Anti-Patterns*. InfoQ. Retrieved 2021, from https://www.infoq.com/articles/Diagnose-Microservice-Performance-Anti-Patterns/.

[52] Carneiro, C., & Schmelmer, T. (2018). *Microservices from day one*.

[53]Aucion, C. (2018). *How Anti-Patterns Can Stifle Microservices Adoption in the Enterprise | Application Performance Monitoring Blog | AppDynamics*. Retrieved 2024, from https://www.appdynamics.com/blog/engineering/how-to-avoid-antipatterns-with-microservices/.

[54] Dietrich, E. (2018). *Top 4 Ways to Make Your Microservices Not Actually Microservices | Scalyr*. Retrieved 2021, from https://www.scalyr.com/blog/top-4-ways-to-make-your-microservices-not-actually-microservices.

[55] Saleh, T. (2016). *Microservices Antipatterns*. InfoQ. Retrieved 2023, from https://www.infoq.com/presentations/cloud-anti-patterns/.

[56] Postman. (2017). Automated Testing. Retrieved from https://www.postman.com/infographics/automated-testing-whitepaper.pdf

[57] Bulaty, W., & Williams, L. (2019). Testing Microservices: an Overview of 12 Useful Techniques. InfoQ. Retrieved 2021, from https://www.infoq.com/articles/twelve-testing-techniques-microservices-intro/

[58] Laban, J. (2020). Why You Need A Microservice Catalog. Retrieved 2024, from https://www.opslevel.com/2020/04/21/why-you-need-a-microservice-catalog/

[59] Simform, (2019). Microservices Testing Strategies, Types & Tools: A Complete Guide. (2019). Retrieved 2020, from https://www.simform.com/microservice-testing-strategies/

[60] Bogard, J. (2017). Avoiding Microservice Megadisasters [Video]. Retrieved 2020, from https://www.youtube.com/watch?v=gfh-VCTwMw8

[61] Liu, L., Tu, Z., He, X., Xu, X., & Wang, Z. (2021, September). An Empirical Study on Underlying Correlations between Runtime Performance Deficiencies and "Bad Antipatterns" of Microservice Systems. In *2021 IEEE International Conference on Web Services (ICWS)* (pp. 751-757). IEEE.

[62] Zhong, C., Huang, H., Zhang, H., & Li, S. (2022). Impacts,

causes, and solutions of architectural antipatterns in microservices: An industrial investigation. Software: Practice and Experience, 52(12), 2574-2597.

[63] Abbott, M. (2022, August 17). Why is my development team so slow? Retrieved May 1, 2024, from https://akfpartners.com/growth-blog/why-is-my-development-team-so-slow

[64] Bottleneck #01: Tech debt. (n.d.). Retrieved May 1, 2024, from https://martinfowler.com/articles/bottlenecks-of-scaleups/01-tech-debt.html

[65] Tighilt, R., Abdellatif, M., Trabelsi, I., Madern, L., Moha, N., & Guéhéneuc, Y. G. (2023). On the maintenance support for microservice-based systems through the specification and the detection of microservice antipatterns. Journal of Systems and Software, 111755.

[66] Fang, H., Cai, Y., Kazman, R., & Lefever, J. (2023, March). Identifying Anti-Patterns in Distributed Systems With Heterogeneous Dependencies. In 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C) (pp. 116-120). IEEE.

[67] Matar, R., & Jahić, J. (2023, March). An Approach for Evaluating the Potential Impact of Anti-Patterns on Microservices Performance. In 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C) (pp. 167-170). IEEE.

[68] SAM, P. D. S. (2023). Principles of Software Architecture Modernization: Delivering Engineering Excellence with the art of fixing microservices, monoliths, and distributed. BPB PUBLICATIONS.

[69] Strauss, A., & Corbin, J. (1990). Basics of qualitative research. Sage publications.

[70] Sharma, T., & Spinellis, D. (2018). A survey on software smells. Journal of Systems and Software, 138, 158- 173.

[71] Kessentini, W., Kessentini, M., Sahraoui, H., Bechikh, S., & Ouni, A. (2014). A cooperative parallel search-based software engineering approach for code-smells detection. *IEEE Transactions on Software Engineering*, *40*(9), 841-861

Date available online: 23-07-2025
Vol. 3, Issue 1 (January – June 2025)

DOI: https://doi.org/10.24312/ucp-jeit.03.01.543

Smart Resource Allocation for Mobile Edge Network in IoT Using Game Theory

Samra Shereen[1], Asif Kabir[1], Syed Mushhad M. Gilani[2], Abdur Rehman Riaz[3], Zahid Mahmood[1]
[1]Department of CS and IT, University of Kotli, Kotli, Azad Jammu and Kashmir, Pakistan
[2]Department of Computer Science, Agriculture University Faisalabad, Pakistan
[3]Department of Computer Science, University of Management and Technology, Sialkot, Pakistan

Corresponding Author: Syed Mushhad M. Gilani (Email:mushhad@uaf.edu.pk)

ABSTRACT-

Emerging from years of research and development, the modern era of computing recognizes the Internet of Things (IoT) as the most empowering technology to connect the digital and real world. IoT has introduced new advancements that are transforming the world, however, it still faces constraints that limit its effectiveness in various application areas, including computing power, resource allocation, reliability, and time consumption. Achieving acceptable latency for task operations on IoT devices necessitates the appropriate allocation of Mobile Edge Caching device computing resources which should be based on task size, delivery, and service latency. It is impossible to handle the billions of data requests originating from a growing number of base stations. This research proposes a mechanism for allocating computing resources and caching to facilitate efficient scheduling in cellular networks. A game theory approach used to model miniaturization problems has been employed in this work. A wireless network system has been analyzed where each node in the system is a participant with its strategies and contributions to achieve the desired performance. The simulation results show that the proposed technique has great potential to improve resource allocation. Each IoT device increases the number of requests handled by the Mobile Edge Computing(MEC) server in the non-cooperative subgame. The proposed system efficiently allocates IoT resources excels in performance and reduces latency.

Index Terms: Internet of Things, Game Theory, Resource Allocation, Mobile Edge Computing

I. INTRODUCTION

Mobile networks have experienced rapid growth over the past decade, offering multimedia, online gaming, and video services. The number of mobile phone users and data traffic has exponentially grown [1]. IoT is one of the most emerging research driven by the widespread adoption of smart technology devices and advancement in communication technologies, including 5G. Wireless networks are expected to have an abundance of devices such as smartphones, portable computing devices, smart sensors, and other growing numbers of physical devices. These devices spread a large volume of data in the network, that's why networks need high-performance computing and a big storage capacity to manage it [2].

However, despite the seemingly unlimited computational capabilities offered by cloud services, this paradigm introduces several challenges like trust, congestion issues, high transmission costs, and prolonged service latency hinder its feasibility in many IoT scenarios that require real-time interaction or mobility. Mobile Edge Computing (MEC) solved the problem of heavy traffic in the network. At the edges of the network processing and caching of data are done. MEC consists of a single-edge server or group of devices that work together to serve mobile users. The MEC is a more

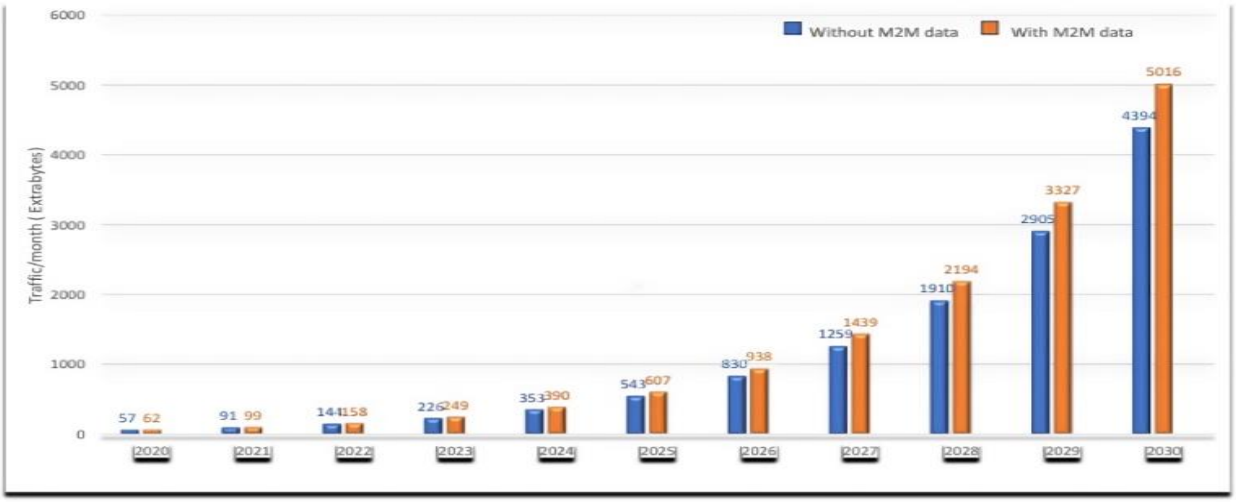

FIGURE 1: Global mobile devices and connection

efficient viable option than the remote cloud because it has much lower network latency. In addition, the MEC will be able to efficiently explore the computing and storage resources available at the edge of the network [3]. Data traffic generated by mobile devices all over the globe is forecast by ITU as shown in Figure 1. According to the studies data traffic rose up to 55% in the decade of 2020 to 2030. The estimated amount of data in 2025 will be 607 Exabytes and 5016 Exabytes in 2030(Source: Cisco).

However, MEC also has some limitations, such as the high cost of implementing and maintaining the architecture, and the enormous pressure that a complex and dynamic IT environment puts on MEC vendors. The rational allocation of computing and network resources to meet the polarization needs of mobile communications under dynamic MEC conditions is extremely challenging today [4]. In connection with the IoT, a neural network search system based on compound learning has been developed. It includes an anomaly classification model at the edge and a distributed learning framework for combining model parameters at the server to create a generic model for all edge regions. This method not only reduces data transmission requirements and increases latency, but also improves user privacy. Although the efficiency was slightly reduced due to the reduced location of the edge, accuracy was achieved by creating models suitable for different scenarios [5].

MEC servers also efficiently plan the resources of the mobile edge networks by joint caching and computing allocation mechanisms. In different environments, base stations (BSs) participate to control the computing space that can inhabit the MEC server to enhance the quality of use of their consumers [6]. Under the condition of mutant MEC, the experimental results show that the proposed Dynamic Reinforcement Learning Resource Allocation (DRLRA) algorithm performs better than the conventional algorithm. 5G wireless networks have been attracting much attention from academia and industry since the last quarter [7]. The biggest challenge is to meet the cost and energy consumption data compared to today's networks. 5G wireless cellular networks will borrow many new technologies to support the growth of wireless transmission services that are yet to be invented [8].

The network segment resource request mediation process improves the instantiation, configuration, and scaling of network segment resource requests when the client-provider relationship of 5G segments is broken. Likewise, the IoT industry can be optimally restructured to accommodate unexpected 5G network traffic. Interestingly, IoT Broker offers different functionalities.

(i) Appropriate selection of IoT Gateway (GW) configured to satisfy downstream order data request or Quality of Services (QoS) parameters (e.g., data accuracy, notification rate).

(ii) Measurement of data activity to cover business fluctuations, notification rate changes of IOT, and changes in QoS parameters during request/subscription delivery.

(iii) Data trading optimization to maximize the efficiency of 5G network applications [9].

Optimizing the allocation of their computing and communication resources, a protocol based on four specialized domains, and developing an energy-efficient design framework to meet the computational silence needs while reducing their overall consumption of energy. The IoT needs to be properly managed, and network performance needs to be improved. In a two-subcaste diverse IoT network with limited network coffers, a distributed Q-learning supplementary power distribution algorithm ensures the fairness of different biases to ensure druggies are treated equally [10].

In recent years, MEC, an efficient computing paradigm, has provided abundant computing resources for IoT. Overall, deploying MEC servers closer to mobile users effectively reduces access latency and the cost of using cloud services. Several mobile applications have been developed to connect the world of things to the Internet. However, to guarantee fair task action latency among IoT devices, calculation resources of MEC units need to be allocated accordingly based on task size whilst considering transmission and service latency. Using deep joint caching and computing learning algorithms, the goal of this study is to make it possible for manipulators to acquire new and challenging skills to solve the issue of resource allocation. How to reasonably allocate computing resources and network resources to meet the needs of mobile devices under the ever-changing conditions of MEC has become an important issue today. To address this problem, we propose an intelligent deep policy based on asset storage and processing learning, which can adaptively identify logs and network assets, reduce typical overtime, and balance the utilization of resources in different MEC conditions.

We yield algorithms from game theory and drive an efficient formula for smart resource allocation. The efficiency and accuracy of our driven formula are validated by using MATLAB. The tool gives us graphs that can show the accuracy of our proposed work with the growth of mobile biases and the improvement of communication skills and cognition, complex, multifaceted, and computationally intensive mobile processors have emerged. Due to limited resources, mobile polarization is increasingly limited. The research article is arranged as follows: Section II discusses the state of art technique from the literature. Our proposed model to describe the system is in Section III. In Section IV we articulate the problems related to the topic. Analysis and discussion of the proposed model is reframed in Section V and the conclusion of our studies is written in Section VI.

II.RELATED WORKS

Resource management algorithms can be distinguished based on their approach to resource allocation such as; Provision is the act of assigning resources to workloads. Allocation is the distribution of resources linking competing loads. Modeling provides a framework that assists in predicting the resources needed for a given

workload [11]. Brokering is the negotiation of resources through an agent. Scheduling is organizing resources, requests, and events in a timetable that links requirements and time intervals for available resources [7]. Resource allocation in mobile edge computing using game theory faces challenges involving different technologies. However, its ability to cover different decision strategies helps to improve the decision process for the allocated time and objectives [8].

A game-theoretical approach to solving the attribution problem of IoT problems. There are three reasons for adopting a game-theoretic approach in this way. First of all, users of your application may have different needs and interests [12]. Game theory has been used successfully in many fields as an effective tool for analyzing the mutual influence of multiple actors acting on their interests. No application user has an incentive to unilaterally deviate, as an incentive-enabled mechanism can be jointly developed in an edge computing environment and is a satisfactory IoT solution. Second, Game seeks to harness the intelligence of individual application users to solve IoT problems in a decentralized way [13], [14]. This can reduce the high search load for centralized optimal solutions. The number of users assigned to the application and the number of edge servers available. Finally, comparing centralized and distributed game-theoretical approaches can quickly find an operating System solution. This allows applications to meet the needs of users and application providers for a low-latency edge computing environment [15]. Additionally, gaming application providers must consider capacity constraints such as CPU, memory, and bandwidth**.** etc. Compared to mega cloud servers data processing is limited in data centers, and edge server's capabilities are typically shared between multiple application vendors. So, the edge server must have enough computing capacity for application users to this edge server [16].

MEC is a promising way to expand the computing competencies of Western Digital (WD). MEC works silently by offloading some or all of the WDs' computing tasks to nearby MEC server access points [17]. Through MEC, small and low-power WDs can offload their computing tasks to access points, and those tasks can always be loaded and computed by embedded MEC servers. However, once the computational tasks are successfully offloaded, the MEC mode can facilitate computationally intensive tasks in real-time through both the original offset computing and the edge computing of the MEC service offloading tasks [18]. All of these are connected to the Internet and produce the low-speed tracking, dimensional, or robotic data that many businesses and end-users routinely require, underscoring the need for online coverage techniques in IoT enterprises. On the other hand, the number of devices on the Internet has recently added a new network factor the future of connectivity to "everything" on the Internet. IoT "Big Data" focuses on the four V's: Velocity, Variability, Volume, and Values. Then we have different data models, produced at different rates, which affect the different volumes of data that are dumped and used in IoT operations. Therefore, it is necessary to consider the latest technologies when handling data. The amount of data generated by mobile and IoT bias has increased significantly. These devices, such as smartphones, wearables, and detectors, have limited computing and energy resources. The decomposition process and inventory of resource-limited bias toward income currently face similar limitations [19].

However, computers are hosted in huge data centers located far from the extreme endpoints. In addition, the increased amount of changed data places a significant burden on network connections. Network functions of IoT service layers can also be virtualized. Several global standards (such as oneM2M) and personal (such as IBM Watson) IoT service sub-box platforms have integrated cloud and IoT to provide scalable IoT services using a slice sub-box. Data centers can perform complex computations and data analysis and are therefore responsible for reusing latency-tolerant services containing large amounts of storage and computing at the head end to improve the processing of edge computing tasks [14]. In particular, the IoT bias sniffs large amounts of data and transfers their services only to edge servers instead of unpacking them directly into balls to reduce the required signaling and corresponding energy consumption in the decision tree [20].

The concept of Multi-Access Edge Computing (MEC), as defined by the European Telecommunications Standards Institute (ETSI), is gaining traction with practical implementations. Given that network slicing and virtualization are fundamental to MEC, this discussion also incorporates the latest advancements in 3GPP technologies. These include mechanisms for slicing IoT service resources, which can be deployed on peripheral boards [21]. A game-theoretic approach called the Edge Resource Allocation (ERA) Game is used to address the challenge of pricing edge server resources owned by multiple stakeholders. This method delivers a solution that satisfies the conditions of a pure Nash equilibrium (PNE) for the ERA problem. By leveraging the ERA Game framework, the ERA algorithm is designed, allowing the system to converge at PNE. Once convergence nears, edge servers are divided into distinct groups, prompting the activation of the ERA algorithm. The algorithm operates concurrently across all edge servers within each cluster. It has been demonstrated that the ERA framework is a viable model, ensuring at least one PNE based on the ERA algorithm [22].

Furthermore, vulnerabilities within network processes are identified and addressed, emphasizing the importance of lifecycle management for resolving such issues. This is crucial for safeguarding digital twins and developing robust network distribution strategies. The study also highlights protective measures to enhance the security of Industrial IoT systems. A key innovation lies in applying game theory to analyze network security risks, offering fresh insights into effectively understanding and mitigating information security vulnerabilities in digital twin networks [23]. Although mobile edge computing can improve the efficiency of Mobile device (MD) applications, the simultaneous transmission of MDs can degrade the

TABLE 1: Comparison of recent game-theoretic approaches for resource allocation in MEC-enabled IoT systems.

Ref	Approach/Method	Key Idea	Strengths	Limitations
[2]	Game-Theoretical Task Allocation	Reward-driven for cognitive IoT	Optimized for user-specific needs	Might be complex for real-time applications
[3]	Game-Theoretical User Allocation	Edge computing environment	Decentralized control, user participation	Scalability under large loads?
[6]	MOACO + RL	Multi-objective optimization with RL	Good performance in IIoT	Training cost of RL
[9]	DRL-Based Resource Management	For Industrial IoT	Self-adaptive, high performance	May suffer from convergence delay
[10]	Caching & Multicast in 5G	Optimizes BS behavior	Better network efficiency	Limited real-time adaptability
[14]	Edge Intelligence & Energy Efficiency	Combines offloading + energy reduction	Scalable, effective for mobile devices	Needs careful balancing
[22]	ERA Game Model	Nash Equilibrium-based pricing	Fast convergence, fairness	Grouping overhead possible
[25], [26]	DDRM Algorithm with DDPG	Solves high-dimensional MDP	Reduced task arrival delays	High training complexity
[30]	DRL-Based MEC Task Scheduling	Optimizes task delay	Dynamic & adaptive	Initial model training required

channel quality. Although the clustering technique is used for wireless data transmission, previous computational decoding studies rarely used the concept of clustering to improve the efficiency of game-theoretic decoding [24].

A brand-new videotape analytics frame for blockchain-enabled Internet of autonomous vehicles (IoAV) with MEC script in which a videotape offloading and resource allocation problem is formulated to reduce the system's idleness and maximize the blockchain system's sale output. The possibility of an incorrect successive interference cancellation (SIC) has been investigated in non-orthogonal multiple access (NOMA) based IoT systems. An energized effectiveness optimization problem has been formulated to pair and allocate the radio resource.

An algorithm dynamic resource management (DDRM) was developed to explain the model for sequential stochastic decision problems (MDP), using Deep Deterministic Policy Gradient (DDPG) to handle, high dimensionality of state and action space. Experimental results showed that DDRM could effectively decrease task arrival rate compared to uniform resource management and random resource management algorithms [25], [26]. The socialization of resource sharing, value creation, user participation, supply-personalization, on-demand use, and demand matching in manufacturing run much more clearly and quickly. Wider applications of these Application management systems are hampered by a lack of open architecture, common specifications and standards, intelligent perception, and internet connectivity for the underlying physical manufacturing resources [27]. A novel process model might use fog computing. It brings cloud services and computing to the end of the network.

The IoT landscape consists of connections between different network anomalies. Virtual machine (VM) origin is far from an isolation scheme, as moving a virtual machine in real-time allows you to move an entire running task to another virtual machine. Based on this approach, Clone Pall and Cloudlet proposed computational offloading to Pall by running mobile polar tasks on remote virtual machines without programming [28], [29]. MEC collaboration in computing and communications is proposed by [30] which uses deep reinforcement learning-based dynamic resources management algorithm to lessen the long-term average delay of tasks to improve the performance of IoT.

The content caching mechanism is used to improve data delivery and its efficiency. A geographical cluster model is design for the retrival of content and algorithms are used to fine delays and transmission cost [31]. Caching methods are cetegroized based on their location, granularity, and coordination mechanisms, highlighting their effects on reducing latency and offloading core networks [32]. Artificial intelligence is impacting every domain in computing, cashing algorithms and techniques are also use machine learning and deep learning [33]. The [34] use deep learning base algorithms to optimizing resource allocation in vehicular networks. The proposed deep reinforcement learning model enables vehicles to act as intelligent agents that dynamically allocate communication resources based on environmental feedback and network conditions.

III. SYSTEM MODEL AND DESCRIPTION

The system architecture in Figure 2 of this paper has four layers: the cloud, the MEC layer, the user, and the IoT device
layer. The IoT device layer includes various gadgets like mobile phones, and smart IoT base environments which contain sensors and actuators that scan the environment and collect raw data. The user layer allows each user to control and process the IoT device's data. The MEC layer receives all the raw data and performs data pre-processing and analysis. The output data is reversed back to the user or to the cloud for more analysis and future use. The user should have an interactive real-time

TABLE 2: Description of symbols used in the equation.

Notation	Description	Notation	Description
BSs	Base station	b_k	Segments width
K	Number of Base stations	λ_k	Index
ECN	Edge Computing Nodes	μ_w	Average Service Rate
d_k	Service delay threshold time	D_r^w	Remaining Execution Time
τ_k^{th}	Quality of service requirements	D_t^w	Delay Time of edge
t_k^{net}	Represents network delay	η_w	Number of Segments
t_k^{comp}	Initial component delay time	λ	Average Arrival Rate

application that shows the data analysis results immediately. The MEC layer consists of a group of edge computing nodes (ECN), each with several low-power computing resources that can store computing devices.

Each BS is connected to an MEC server which acts as a small data center. The BS and the MEC server are in the same network and can cache and store content locally. This reduces the service latency and network congestion as the content is closer to the end users. Video streaming is one of the main applications that take advantage of this technology.

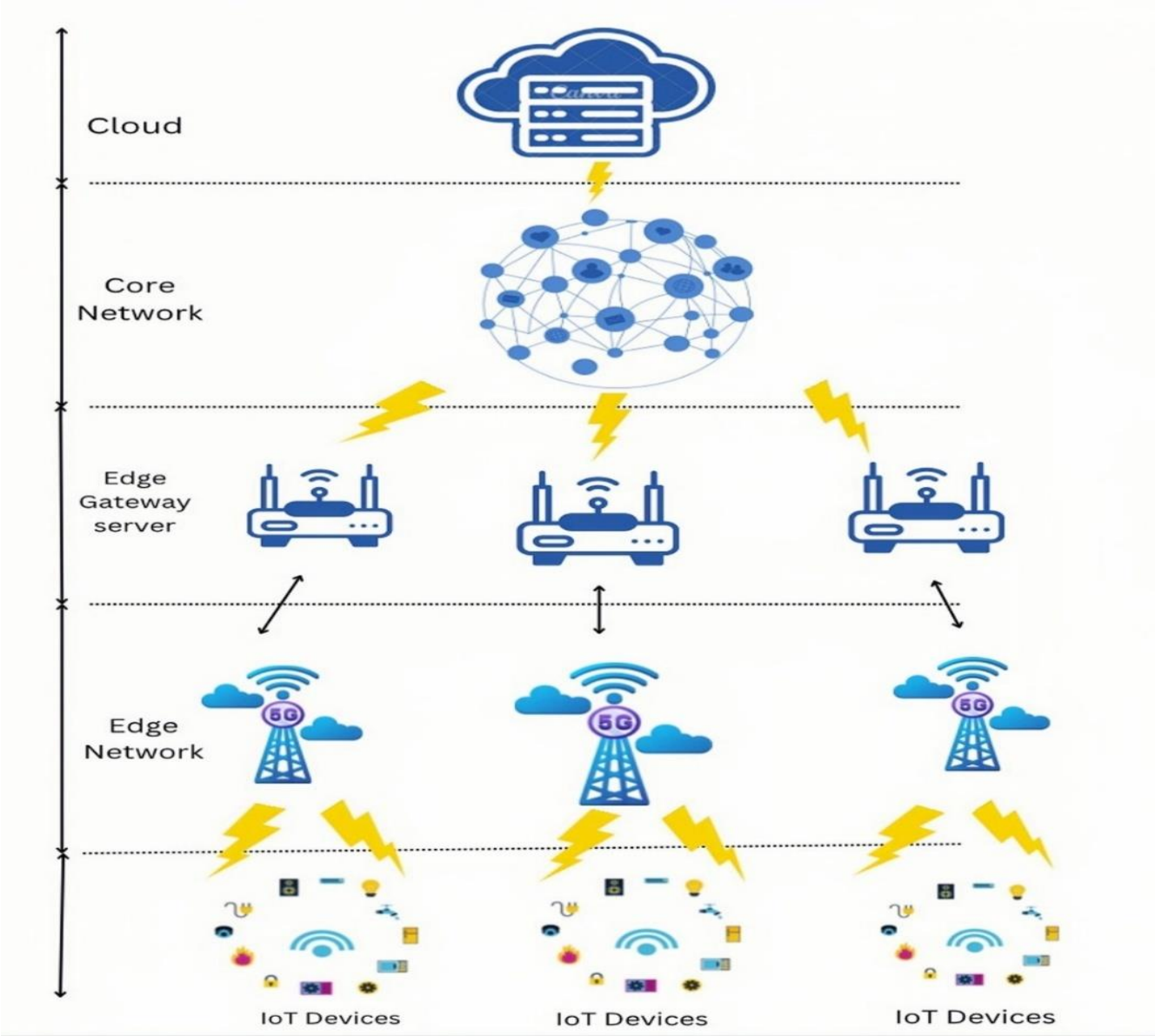

FIGURE 2: System model with four layers IoT Networks

When an end-user requests for a video stream with a given bit rate in which the MEC server works, it responds to the request first then the requested video stream is saved in the MEC server storage. Otherwise, it is requested to the cloud, which will take time increase the latency in response time, and use network supply to provide the desired video. Additionally, if the video is not cached at the requested bitrate and the video is cached at a higher bitrate, the request can also be satisfied by transcoding on the MEC server.

IV. PROBLEM FORMULATION

The problem formulation outlines a resource allocation strategy in IoT networks supported by MEC. To model computing capacity, the system uses Computing Resource Blocks (CRBs), while considering base station competition and latency constraints. Key parameters, including service delay thresholds, network delay, and the computing power of Edge Computing Nodes (ECNs), are clearly defined and incorporated into the model. Transmission latency is derived using Shannon's equation, accounting for time-varying channel conditions, and request patterns follow a Poisson distribution to reflect realistic traffic behavior. The formulation adopts queuing theory to estimate execution delays and system load, introducing metrics such as remaining service time and task queue length. The optimization challenge is initially non-convex but is shown to be convex under certain parameter conditions, which are explicitly stated. A game-theoretic model specifically, a Stackelberg game is employed to model the interaction between the MEC server and base stations. The proposed solution uses the Newton–Raphson method in an iterative algorithm, with each step of the algorithm, including initialization, sorting, and convergence criteria, thoroughly described to ensure reproducibility. The rationale behind selecting this model lies in its ability to balance fairness, utility maximization, and computational feasibility in a distributed network environment.

The MEC server can distribute Computing resources in units called CRBs. Each CBR can offer computing services at a rate of μ. Suppose that the storage capacity of MEC server is set as Q^s and CRB named as Q^c. Moreover, for improving the quality of service the range of M is M ∈ {1, 2, ..., M}. Which competes for the partial resources of BSs and MEC server of its users. User will communicate with their respective BSs through devices. If we have M number of users then λ_m is for mth BS users request. There are two categories for requesting the server one is for video service and other is for data service. We know that, 0 ≤ η ≤ 1, this is the presentation of proportional relation of video and service request. Therefore, the arrival rate of the entire video service request to the BS can be expressed as follows $\eta\lambda_m$.

Suppose that there are entirely K BSs identified as d_k and k ∈ {1, 2, . . ., K} and W, ECNs assigned by $f_w$here w ∈ {1, 2, . . ., W}. Other BSs have dissimilar calculation criteria that can be calculated using service lag. For example, particular BSs may select the lowest service delay at the cost of higher costs, while others may require the lowest cost at the cost of longer computations. The service delay threshold time d_k, required to meet the quality-of-service requirement is denoted by τ_k^{th}. In other words, the total delay in serving in the segment d_k, given as t_k, must satisfy the requirement $t_k \le \tau_k^{th}$. Now, the total delay for serving of section from BS to d_k, consists of both components,

$$\boldsymbol{t_k = t_k^{net} + t_k^{comp}} \quad \textbf{(1)}$$

Here the initial component t_k^{net} represents the network delay t_k^{comp}, which indicates the aggregate adjournment of together the delaying time and the service time.

The wireless channel involves the mobile edge caching and BS as a credible time-varying network, as a Finite Markov Channel (FSMC). The arriving SNR is shown γ_k^w, whose transition follows a Markov process. Therefore, the network latency t_k^{net} can expressed according to the Shannon equation, as, ${}^{o_k}/_{b_k}\ \log(1+\gamma_k^w)$ where $o_k$data represents the dimension for the segment, and b_k is the width.

The data segments of BSS d_k follow a Poisson division with an index λ_k, k = {1, 2, . , . ., K}. For edge computation, each ECN is considered to have different computing power and ECN f_w is assumed to be able to run computation service with an average service rate of μ_w. It is assumed that one data segment from BSS d_k is configured to serve by ECN f_w presenting smart contract. The calculation time of arc representation segments can be divided into two parts: The delay timef_w as part of the smart contract. The calculation time of arc representation segments can be divided into two parts: The delay time D_q^w and D_t^w, respectively. Therefore, the entire computation delay time of the edget_k^{comp}can be expressed as

$$\boldsymbol{t_k^{comp} = D_q^w + D_t^w} \quad \textbf{(1.5)}$$

The average computation time of the ECN f_w server data segment for the CPU can be calculated from the average service frequency, which is 1/μw. Then, equation (2) can be updated as follows equation (3) can be written as

$$\boldsymbol{t_k^{comp} = D_q^w + \frac{1}{\mu_w}} \quad \textbf{(2)}$$

$$\boldsymbol{t_k^{comp} = D_r^w + \frac{n_w}{\mu_w}} \quad \textbf{(3)}$$

where D_r^w is the remaining execution time of the segment on the server n_w show how many segments in the queue moving to the next segment d_k from the BSs is on time. For doing this job first-come, first-served technique is used, means each ECN is counted one shared in one time in the beginning of the queue. For making ECN load free we send the data to the cloud or base station when computing is complete for one of it. For convenience, the average time for processing is approx. D_r^w, which is shown below:

$$\boldsymbol{D_r^w = \frac{1}{2}\lambda\frac{1}{\mu_{w^2}}} \quad \textbf{(4)}$$

The average arrival rate is the shown as λ and f_w. So, the entire delay time of edge calculation can be done on it as

$$\boldsymbol{t_k^{comp} = \frac{1}{2}\lambda\frac{1}{\mu_{w^2}} + \frac{n_w}{\mu_w}} \quad \textbf{(5)}$$

Currently, the expression of the total serving delay f_w for one BSS segment d_k by ECN can be expressed as

$$\boldsymbol{t_k} = \frac{o_k}{b_{k\log(1+\gamma_k^w)}}\frac{1}{2}\boldsymbol{\lambda}\frac{1}{\mu_{w^2}} + \frac{n_w}{\mu_w} \leq \boldsymbol{\tau_k^{th}} \qquad \textbf{(6)}$$

It shows that the time spent on the validation activity is excluded from the absolute value of total time. After completing the calculation, the calculation is completed, the calculation result should be sent back to ECN f_w or stored in the cloud as soon as the calculation is finished. If the calculation results cannot be verified later ECN f_w tokens will not be received. But a certain percentage of the deposit will be deducted and BSS will be returned. However, a percentage of the deposit will be deducted, and BSs d_k

TABLE 3: Description of symbols used in the theorem.

Notation	Description	Notation	Description
p^c	Service Price	α_m^c	Weight factors of computing resource
$q_m^{c^*}$	Amount of CRB	β_m^c	Utility function of BSs
CRB	Computer Resource Block	α_m^s	Weight factors of caching resource
q_m^s	Size Storage Capacity	β_m^c	Utility function of BSs
p^c	Caching price	o_k	Data segments dimensions
U_m^c	Quasi-Concave Function	$B_m(p^c)$	Best Computation Price

According to the modeled architecture of the network consider which games to allocate computing resources to. Consider the optimization problem q_m^{c*} as follows.

Theorem: if an MEC server advertises its service price p^c, the computer resource allocation model is likely to result optimally in the number of her CRBs with her B_m^c, denoted by q_m^{c*} there is,

$$\boldsymbol{q_m^{c*}} = \left[(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c p^c}} - \mathbf{1}\right)\right] \qquad \textbf{(7)}$$

with an $[\,] \triangleq \text{maximum}(\cdot, 0)$.

Proof: From the utilization of function B_m^c, the first increase U_m^c with respect to q_m^c can be,

$$\frac{\partial U_m^c}{\partial q_m^c} = \frac{\alpha_m^c \mu}{[1+(q_m^c/(t_{th}-\theta d_m))]} - \boldsymbol{\beta_m^c \rho^c} \qquad \textbf{(8)}$$

also, the alternate outgrowth in it q_m^c is

$$\frac{\partial U_m^c}{\partial q_m^c} = -\frac{2\,\alpha_m^c \mu}{[1+(q_m^c/(t_{th}-\theta d_m))]3\,(t_{th}-\theta d_m)} \qquad \textbf{(9)}$$

concerning as $\partial^2 U_m^c / \partial q_m^{c^2} < 0, \forall m \epsilon M$. therefore, U_m^c is a unique function concerning q_m^c .

its highest, i.e.,

$$\boldsymbol{q_m^{c*}} = \left[(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c p^c}} - \mathbf{1}\right)\right] \qquad \textbf{(10)}$$

thus, the responses of the BSs, we adjust that the problem for the MEC server is still extreme, thus, for the m[th] BS, the index function to show

$$\sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c p^c}} - \mathbf{1}\right) \leq \boldsymbol{Q^c}$$

(11)

$$\boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left[\sqrt{\frac{\alpha_m^c}{\beta_m^c p^c}} - \mathbf{1}\right.$$

m[th] value of participates in the game.

$$\boldsymbol{y_m} = \begin{cases} \mathbf{1}, & \boldsymbol{p^c} < \frac{\alpha_m^c \mu}{\beta_m^c} \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \qquad \textbf{(12)}$$

Else (6) With the below index function for the Base station, optimization equation (5) is reformulated as

$$\max \boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M} \boldsymbol{y_m}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c p^c}} - \mathbf{1}\right)$$

(13)

$$\max \boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M} \boldsymbol{y_m}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c p^c}} - \mathbf{1}\right) \leq \boldsymbol{Q^c} \qquad \boldsymbol{y_m} \in \{\mathbf{0}, \mathbf{1}\}$$

It's egregious that the below problem is because of the index, so it's not a convex problem. Nevertheless, it is not hard to prove that problem (7) is convex for a given index vector y. Equation (7) therefore assumes that Q^c is sufficient for entire BSs to participate to the game. As an outcome, all BS pointers are equivalent to 1.

$$\boldsymbol{p^c} < \left(\frac{\alpha_m^c \mu}{\beta_m^c p^c}\right), \forall \boldsymbol{m} \in \boldsymbol{M}, \qquad \textbf{(13.1)}$$

In this statement, the problem is curved and optimal. For solving this problem given below hypothesis could be follow. The best result of equation (7) pointers

(i.e. $y_m = 1, \forall m \epsilon M$) is given by

$$\boldsymbol{p^{c*}} = \mathbf{max}\{\boldsymbol{B_m}(\boldsymbol{p^c}), \boldsymbol{p^{c,LB}}\} \qquad \textbf{(14)}$$

where $B_M(p^c)$ satisfied

$$\boldsymbol{B_M}(\boldsymbol{p^c}) = \arg \max_{\boldsymbol{p^c}} \boldsymbol{U_{MEC}^c}, \quad \forall \boldsymbol{m} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{M} \qquad \textbf{(15)}$$

$$\boldsymbol{p^{c,LB}} = \left\lceil \frac{\sum_{m=1}^{M}(t_{th}-\theta d_m)\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c}}}{\sum_{m=1}^{M}(t_{th}-\theta d_m)+Q^c} \right\rceil \qquad \textbf{(16)}$$

Proof

In expressed problem (7), we take the first outgrowth of

$$\boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left[\sqrt{\frac{\alpha_m^c}{\beta_m^c p^c}} - \mathbf{1}\right. \qquad \textbf{(17)}$$

$$\frac{\partial U_{MEC}^c}{\partial p^c} = \sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\frac{1}{2}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(1/2)}}}\right) + \sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\,\boldsymbol{e^c}\left(\frac{1}{2}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(3/2)}}}\right) - \sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m}) \qquad \textbf{(18)}$$

also, the alternate outgrowth with respect to q_m^c is

$$\frac{\partial^2 U_{MEC}^c}{\partial p^{c2}} = -\sum_{m=1}^{M}(t_{th} - \theta d_m)\left(\frac{1}{4}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(3/2)}}}\right) - \sum_{m=1}^{M}(t_{th} - \theta d_m)\, e^c\left(\frac{3}{4}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(3/2)}}}\right) \quad (19)$$

It's obviously seen that $\left(\frac{\partial^2 U_{MEC}^c}{\partial p^{c2}}\right) < 0, \forall p^c > 0$. Thus, $\frac{\partial U_{MEC}^c}{\partial p^c}$, this is a monotonically reducing function in the interval 0, ∞. Also, as the equation shows,

$$\lim_{p^c \to \infty} \frac{\partial U_{MEC}^c}{\partial p^c} = -\sum_{m=1}^{M}(t_{th} - \theta d_m) < 0 \quad \text{and} \quad \lim_{p^c \to 0} \frac{\partial U_{MEC}^c}{\partial p^c} = +\infty > 0 \quad (20)$$

is always decided. therefore, U_{MEC}^c has a definite maximum value. is using $B_M(p^c)$ to indicate the satisfied price (13). Based on the following analysis, an algorithm named Newton-Raphson is proposed by our model which obtain the upcoming value $B_M(p^c)$.

The total number of CRBs assigned to each base station must be in the capacity of the MEC Server's limited processing resources, the computing cost must satisfy inequality (8) price calculation.

$$p^{LB} = \left[\frac{\sum_{m=1}^{M}(t_{th} - \theta d_m)\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c}}}{\sum_{m=1}^{M}(t_{th} - \theta d_m) + Q^c}\right]^2 \quad (21)$$

Therefore, the optimal hidden cost determined by the MEC Server (9) can be attained. Judging from the given results, our model is able to work in general situations (7). Assuming all BSs are ordered, the algorithm also gives the optimal result for the problem.

Proposed Algorithm

Algorithm: An iterative algorithm based on the Newton–Raphson method $B_M(p^c)$

Step 1: Set K=M and $y_m = 1, \forall m \in M$

Step 2: Sort the K BSs according to $\frac{\alpha_m^c \mu}{\beta_m^c}$

i.e., $\left(\frac{\alpha_1^c \mu}{\beta_1^c} > \cdots > \frac{\alpha_{M-1}^c \mu}{\beta_{M-1}^c} > \frac{\alpha_M^c \mu}{\beta_M^c}\right)$

Step 3: Compute $B_M(p^c)$ based on $p^{c,k+1} = p^{c,k} - \frac{\partial U_{MEC}^c}{\partial p^c} \Big/ \frac{\partial^2 U_{MEC}^c}{\partial p^{c2}}$

$B_M(p^c) = p^{c,k}$

$$p^{LB} = \left[\frac{\sum_{m=1}^{M}(t_{th} - \theta d_m)\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c}}}{\sum_{m=1}^{M}(t_{th} - \theta d_m) + Q^c}\right]^2$$

Step 4: $p^{c,best} = \max\{B_M(p^c), p^{c,LB}\}$

Step 5: Compare the $p^{c,best}$ with $\frac{\alpha_K^c \mu}{\beta_K^c}$:

If $p^{c,best} < \frac{\alpha_K^c \mu}{\beta_K^c}$ then

go to step 3.

Step 6: $p^{c,Optimal} = p^{c,best}$

The SE realized by the proposed system enables well-balanced utility usage among the mobile edge caching server and numerous IoT base stations.

V. RESULTS AND ANALYSIS

This section presents the results of our proposed model. We simulate our outcomes on MATLAB that give a rich and precise yields. While doing simulated the IoT-based scenario without explicit clarification, we can observe that maximizing the capacity of the MEC server storage for more benefits. As it grows, the utility increases, until reaching a certain point where further increases do not bring any extra advantage. The simulation was conducted in an environment consisting of 25 BSs and with a calculated computation capacity of 50 all having the same Zipf distribution characteristic (τ at 0.5). Every base station's request arrival rate is randomly set to average 10 ms 1 with video service requests comprising half of this rate. On average, each BS can cache 500 videos. Additionally, as per [29], Each CRB has a service rate of 0.1 minutes per second, but all 25 BSs have a delay tolerance of 60 ms.

The storage capacity has a major influence on the cache price. When it is limited, competition among BSs is so furious that the cost of cache is extremely high. However, as the storage capacity increases, the cache price gradually decreases until reaching an equilibrium point (i.e., when pLB ≤ e s / (1 − τ)). Upon attaining this point, the cache fee no longer relies on storage capacity and remains constant.

As indicated in Figure 3, given a storage size of 50, computation power has a positive effect on MEC server utility, raising initially and then stabilizing to a certain value. This occurs as more CRBs are available for allocation to the BSs. Furthermore, with an increased number of BSs and computation level remaining equal, the heightened rivalry between BSs causes converged value to grow higher.

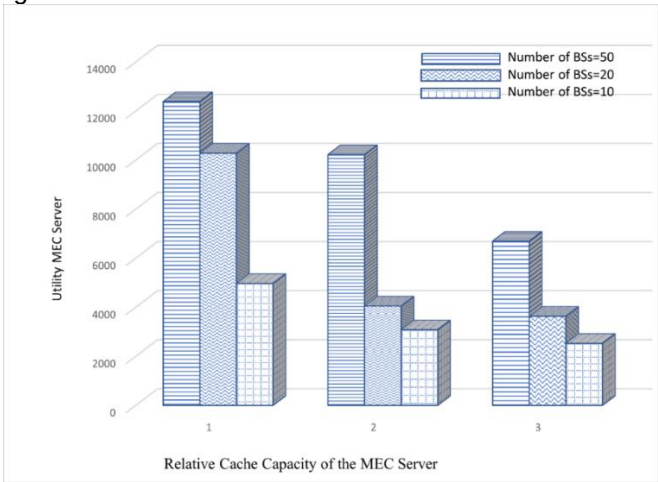

FIGURE 3: Influence of cache capacity on utility MEC server for different sets of BSs

This research evaluates the performance of the mobile edge caching server against its capacity by presenting simulation results. The transmission distance between it and its base stations is a randomly generated number

between 0 and 10 km, while the rent follows a random uniform distribution between 1 and 25. Furthermore, the average weight factors for the server and CRBs are α / β = 50/0.2 and α s/ βs= 500/0.1, respectively. Figure 4 shows our findings.

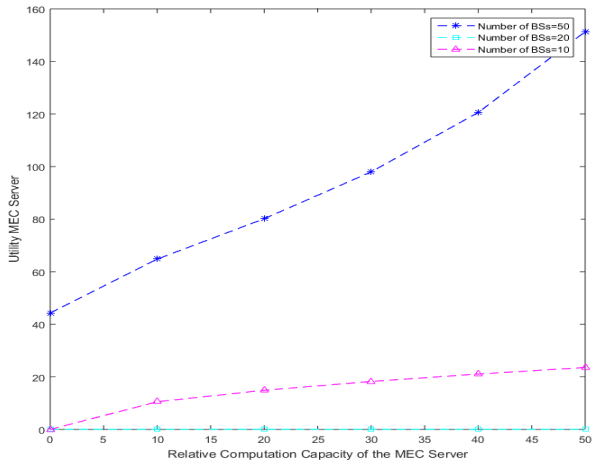

FIGURE 4: Variation in Utility MEC Server with computation capacity

The game-based scheme proposed by Stackelberg is compared with other two methods.

(i) The first one is the YM (Yield Management) Approach which sees the MEC server offering discounts on available resources increments of 30 regarding cache and computation capacity comes with a 5% discount on the current price.

(ii) Second is the greedy scheme, in which MEC acts like a monopolist and enhances overly expensive prices.

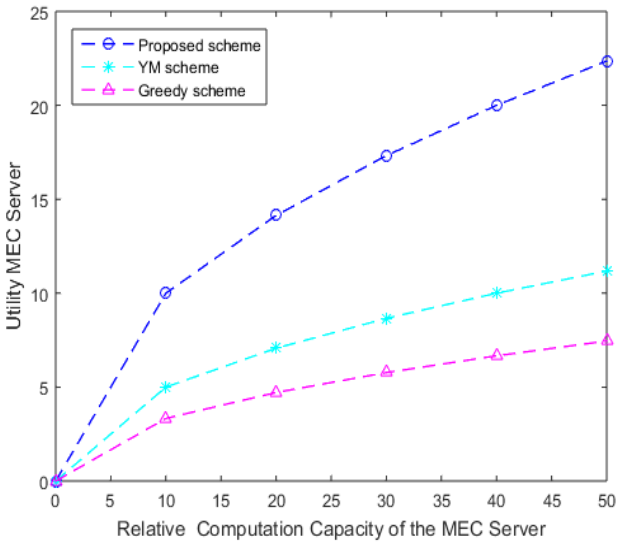

FIGURE 5: Relative computation capacity behaviour of MEC Server under diverse patterns

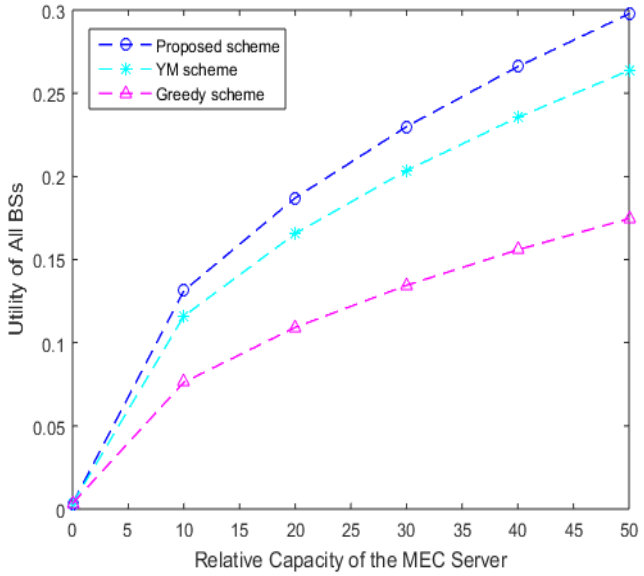

FIGURE 6: BSs efficacy with the capability of MEC server

According to Figures 5 and 6, the utility of MEC servers and BSs are determined by three pricing models concerning resource capacity, respectively. A greedy system allows MEC servers to catch up with Stackelberg's game-based system, as MEC servers always offer the best prices. Figure 6 proves that the SE completed by the suggested system can well balance utility among the MEC servers and numerous IoT base stations.

Figure 7 shows the utility estimation of the MEC server when the service speed μ alters from 0.1 to 1 with a step size of 0.1. Utility increases by μ. Here's why a larger value of μ allows the MEC server to handle more requests directly instead of forwarding them to withdrawn servers within an acceptable service delay for IoT. Therefore, MEC servers can generate more revenue from base station requests.

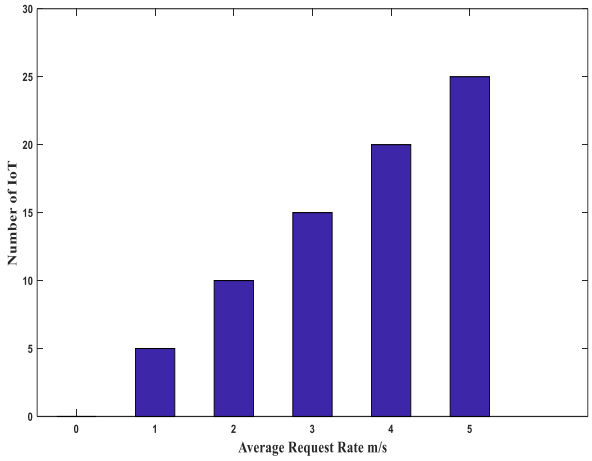

FIGURE 7: Number of IoTs versus the request rate

This research evaluates the performance of the MEC server against its capacity by presenting results. The transmission distance between it and its base stations is a randomly generated number between 0 and 10 km, while the rent follows a random uniform distribution between 1 and 50. Furthermore, the average weight factors for the

server and CRBs are α / β = 50/0.2 and αs/ βs= 500/0.1, respectively.

VI. CONCLUSION

Our research proposes a framework named MEC for resource allocation servers and base station connections to ensure efficient resource planning of IoT cellular networks. The proposed algorithm significantly increases the system efficiency and reduces the response time. The simulation results show the effectiveness of the proposed system. In addition, the originality and equilibrium connection of the Stackelberg game and the reverse induction system are proposed as a solution to the resource allocation problem. For future work, we encourage you to consider efficient computation offload strategies for cross-IoT collaboration. Our future work integrates with neural networks for better analysis of the system and adds predictive capabilities. Adding experimental tests makes it easier for users. Multiple optimization objects are another promising direction for improving overall network performance and are also considered as a new research direction for future research.

REFERENCES

[1] Annual, C., & Report, I. (2018). *White paper Cisco public*.

[2] Rahman, T. F., Pilloni, V., & Atzori, L. (2019). Application Task Allocation in Cognitive IoT: A Reward-Driven Game Theoretical Approach. *IEEE Transactions on Wireless Communications*, *18*(12), 5571–5583. https://doi.org/10.1109/TWC.2019.2937523

[3] He, Q., Cui, G., Zhang, X., Chen, F., Deng, S., Jin, H., Li, Y., & Yang, Y. (2020). A game-theoretical approach for user allocation in edge computing environment. *IEEE Transactions on Parallel and Distributed Systems*, *31*(3), 515–529. https://doi.org/10.1109/TPDS.2019.2938944

[4] Kim, S., Cai, H., Hua, C., Gu, P., Xu, W., & Park, J. (2020). Collaborative Anomaly Detection for Internet of Things based on Federated Learning. *2020 IEEE/CIC International Conference on Communications in China, ICCC 2020*, 623–628. https://doi.org/10.1109/ICCC49849.2020.9238913

[5] Tang, Q., Xie, R., Huang, T., & Liu, Y. (2019). Jointly caching and computation resource allocation for mobile edge networks. *IET Networks*, *8*(5), 329–338. https://doi.org/10.1049/iet-net.2018.5111

[6] Vimal, S., Khari, M., Dey, N., Crespo, R. G., & Harold Robinson, Y. (2020). Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT. *Computer Communications*, *151*, 355–364. https://doi.org/10.1016/j.comcom.2020.01.018

[7] Zhang, J., Hu, X., Ning, Z., Ngai, E. C. H., Zhou, L., Wei, J., Cheng, J., Hu, B., & Leung, V. C. M. (2019). Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching. *IEEE Internet of Things Journal*, *6*(3), 4283–4294. https://doi.org/10.1109/JIOT.2018.2875917

[8] Ning, Z., Dong, P., Kong, X., & Xia, F. (2019). A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things. *IEEE Internet of Things Journal*, *6*(3), 4804–4814. https://doi.org/10.1109/JIOT.2018.2868616

[9] Chen, Y., Liu, Z., Zhang, Y., Wu, Y., Chen, X., & Zhao, L. (2021). Deep Reinforcement Learning-Based Dynamic Resource Management for Mobile Edge Computing in Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, *17*(7), 4925–4934. https://doi.org/10.1109/TII.2020.3028963

[10] Poularakis, K., Iosifidis, G., Sourlas, V., & Tassiulas, L. (2016). Exploiting Caching and Multicast for 5G Wireless Networks. *IEEE Transactions on Wireless Communications*, *15*(4), 2995–3007. https://doi.org/10.1109/TWC.2016.2514418

[11] Bandopadhyay, A., Mishra, V., Swain, S., Chatterjee, K., Dey, S., Mallik, S., ... & Soufiene, B. O. (2024). EdgeMatch: A Smart Approach for Scheduling IoT-Edge Tasks With Multiple Criteria Using Game Theory. IEEE Access.

[12] Hu, P., Ning, H., Qiu, T., Zhang, Y., & Luo, X. (2017). Fog computing based face identification and resolution scheme in internet of things. *IEEE Transactions on Industrial Informatics*, *13*(4), 1910–1920. https://doi.org/10.1109/TII.2016.2607178

[13] Antonius, F. (2024). Efficient resource allocation through CNN-game theory based network slicing recognition for next-generation networks. Journal of Engineering Research.

[14] Dai, Y., Zhang, K., Maharjan, S., & Zhang, Y. (2020). Edge Intelligence for Energy-Efficient Computation Offloading and Resource Allocation in 5G beyond. *IEEE Transactions on Vehicular Technology*, *69*(10), 12175–12186. https://doi.org/10.1109/TVT.2020.3013990

[15] Cuervo, E., Wolman, A., Cox, L. P., Lebeck, K., Razeen, A., Saroiu, S., & Musuvathi, M. (2015). Kahawai: High-quality mobile gaming using GPU offload. *MobiSys 2015 - Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 121–135. https://doi.org/10.1145/2742647.2742657

[16] Chen, W., Wang, D., & Li, K. (2019). Multi-User Multi-Task Computation Offloading in Green Mobile Edge Cloud Computing. *IEEE Transactions on Services Computing*, *12*(5), 726–738. https://doi.org/10.1109/TSC.2018.2826544

[17] Chu, Z., Xiao, P., Shojafar, M., Mi, D., Mao, J., & Hao, W. (2021). Intelligent Reflecting Surface Assisted Mobile Edge Computing for Internet of Things. *IEEE Wireless Communications Letters*, *10*(3), 619–623. https://doi.org/10.1109/LWC.2020.3040607

[18] Feng, H., Chen, D., Lv, H., & Lv, Z. (2023). Game theory in network security for digital twins in industry. *Digital Communications and Networks*. https://doi.org/10.1016/j.dcan.2023.01.004

[19] Huang, Y. Y., & Wang, P. C. (2023). Computation Offloading and User-Clustering Game in Multi-Channel Cellular Networks for Mobile Edge Computing. *Sensors*, *23*(3). https://doi.org/10.3390/s23031155

[20] Ibrahim, A. M., Chen, Z., Eljailany, H. A., Yu, G., Ipaye, A. A., Abouda, K. A., & Idress, W. M. (2024). Advancing 6G IoT networks: Willow Catkin packet transmission scheduling with AI and Bayesian game-theoretic approach-based resource allocation. Internet of Things, 101119.

[21] Kim, Y., Song, C., Han, H., Jung, H., & Kang, S. (2020). Collaborative Task Scheduling for IoT-Assisted Edge Computing. *IEEE Access*, *8*, 216593–216606. https://doi.org/10.1109/ACCESS.2020.3041872

[22] Kumar, S., Gupta, R., Lakshmanan, K., & Maurya, V. (2022). A Game-Theoretic Approach for Increasing Resource Utilization in Edge Computing Enabled Internet of Things. *IEEE Access*, *10*, 57974–57989. https://doi.org/10.1109/ACCESS.2022.3175850

[23] Li, X., Liu, Y., Ji, H., Zhang, H., & Leung, V. C. M. (2019). Optimizing resources allocation for fog computing-based internet of things networks. *IEEE Access*, *7*, 64907–64922. https://doi.org/10.1109/ACCESS.2019.2917557

[24] Premsankar, G., Di Francesco, M., & Taleb, T. (2018a). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

[25] Wang, T., Qiu, L., Sangaiah, A. K., Liu, A., Bhuiyan, M. Z. A., & Ma, Y. (2020). Edge-Computing-Based Trustworthy Data Collection Model in the Internet of Things. *IEEE Internet of*

Things Journal, *7*(5), 4218–4227. https://doi.org/10.1109/JIOT.2020.2966870

[26] Xue, H., Huang, B., Qin, M., Zhou, H., & Yang, H. (2020). Edge Computing for Internet of Things: A Survey. *2020 International Conferences on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 755–760. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00130

[27] Premsankar, G., Di Francesco, M., & Taleb, T. (2018b). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

[28] Zhang, G., Zhang, S., Zhang, W., Shen, Z., & Wang, L. (2021). Joint Service Caching, Computation Offloading and Resource Allocation in Mobile Edge Computing Systems. *IEEE Transactions on Wireless Communications*, *20*(8), 5288–5300. https://doi.org/10.1109/TWC.2021.3066650

[29] Zhang, H., Xiao, Y., Bu, S., Niyato, D., Yu, F. R., & Han, Z. (2017). Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching. *IEEE Internet of Things Journal*, *4*(5), 1204–1215. https://doi.org/10.1109/JIOT.2017.2688925

[30] Zhang, K., Leng, S., He, Y., Maharjan, S., & Zhang, Y. (2018). Mobile Edge Computing and Networking for Green and Low-Latency Internet of Things. *IEEE Communications Magazine*, *56*(5), 39–45. https://doi.org/10.1109/MCOM.2018.1700882

[31] Kabir, A. (2018). Cooperative Content Caching and Distribution in Dense Networks. KSII Transactions on Internet & Information Systems, 12(11).

[32] Kabir, A., Rehman, G., Gilani, S. M., Kitindi, E. J., Ul Abidin Jaffri, Z., & Abbasi, K. M. (2020). The role of caching in next generation cellular networks: A survey and research outlook. Transactions on Emerging Telecommunications Technologies, 31(2), e3702.[1][w]

[33] Kumar, Y., Marchena, J., Awlla, A. H., Li, J. J., & Abdalla, H. B. (2024). The AI-Powered Evolution of Big Data. Applied Sciences, 14(22), 10176. https://doi.org/10.3390/app142210176

[34] Ergün, Serap. "Resource allocation optimization for effective vehicle network communications using multi-agent deep reinforcement learning." *Journal of Dynamics and Games* 12.2 (2025): 134-156.

Date available online: 23-07-2025
Vol. 3, Issue 1 (January – June 2025)

DOI: https://doi.org/10.24312/ucp-jeit.03.01.591

RUPT: An Extension to Traditional Compilers in C++ to Support Programming in Native Language

Muhammad Ishtiaq[1,3*], **Maryam Gulzar**[2], **Muhammad Farhat Ullah.**[1]
[1]School of Software Technology, Dalian University of Technology, Dalian, Liaoning, China
[2]Independent Researcher, Dalian, Liaoning, China
[3]Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China

* Corresponding Author: Muhammad Ishtiaq (E-mail: ishtiaqrai8@gmail.com).

ABSTRACT
The medium of instruction has a significant impact on effective communication and comprehension. Most literature is in English, but presenting information in persons´ native language improves comprehension. In computer science, source code of programming languages is written in the English language, whereas endemic language has its own impact. To address this gap, this study has rendered a framework, "Roman Urdu Programming Translator" (RUPT), that will be used to translate a program coded in Roman Urdu or Hindi into a proportionate C++ program. RUPT acts as a layer above the C++ compiler, allowing programmers to write code using Roman Urdu keywords, which it translates into standard C++ for compilation and execution. The special set of Roman Urdu keywords includes, e.g., "keyboardSay (ks)" instead of "cin", "screenKiTaraf (skt)" instead of "cout", "klea" instead of "for" etc. RUPT replaces added Roman Urdu and Hindi keywords with equivalent C++ keywords to produce valid C++ code. It is only composed of the lexical analysis phase. State of the art has increased the understanding and learning rate of novel users towards the field of computer science.

INDEX TERMS C++, Compiler, Lexical Analysis, Native Language, Roman Urdu, RUPT

I. INTRODUCTION

An accelerator is a piece of hardware or software that has as its primary objective to improve the overall performance of the computer. A variety of accelerators are available to aid in improving the efficiency of various parts of a computer's operation. Due to their high performance and energy economy, numerous specialized Deep Neural Network (DNN) accelerators have also recently become more and more popular [1]. They have been implemented in servers, data centers, and pervasive computing [2], [3]. These accelerators concentrate on particular customization for DNN computations. DNNs are typically represented as computation graphs, where nodes stand for fundamental operations (like operators, such as convolution, pooling, and activation), and edges stand for the data that these operators consume or produce. To accelerate computation, these operators can be offloaded to accelerators [1]. Accelerators facilitate making efficient use of computer resources, but we need to grease the wheels for human beings and motivate them to dive into the field of computer science. Learning the native language is very important, as it has a great impact on the education of children. It has proved its unique importance as a key factor in getting awareness of new developments in studies and success in future life. The relationship between conscious self-regulation and executive functions, two groups of regulatory predictors, and academic performance in the native language has been discovered by structural equation modeling [4]. By knowing the importance of native language in academics, in this paper, work has been done to provide a platform that will support writing source code of programming language in native language. Programming languages use English words to code. Roman Urdu is the name for writing the Urdu language in Roman characters [5]. A recently developed language in South Asia is called Roman Urdu. Romanized Urdu deviates from the rules of the Urdu language. However, many internet users utilize this language to communicate their views and ideas on a variety of topics [5], [6]. Daud [7] estimates that 300 million people use the Urdu language worldwide. Additionally, there are roughly 500 million native Hindi speakers, according to Kunchukuttan et al. [8]. The majority of them are literate in Roman Urdu. As a result, we can estimate that there are about 800 million Roman Urdu speakers [10]. Roman Urdu, which appears in the last column of Figure 1, is an example of three sentences that both Urdu and Hindi speakers may understand.

Sr. No.	English Phrase	Urdu Phrase	Hindi Phrase	Roman Urdu Phrase
1	I play	میں کھیلتا ہوں	मैं खेलता हूँ	main khelata hoon
2	I love to read book	مجھے کتاب پڑھنا اچھا لگتا ہے	मुझे किताब पढ़ना अच्छा लगता है	mujhe kitaab padhana achchha lagata hai
3	I am working	میں کام کر رہا ہوں	मैं काम कर रहा हूं	main kaam kar raha hoon

FIGURE 1: Three phrases are compared using several foreign languages

Zain et al. [19] introduced RU-OLD, a hate speech detection model for Roman Urdu that integrates deep learning, transfer learning, and hyperparameter optimization. Their study highlights the linguistic challenges associated with Roman Urdu, particularly the need for effective tokenization strategies—aspect also addressed in the present work. In this context, we propose RUPT (Roman Urdu Programming Translator), a platform designed to enable programmers to write C++ code using Roman Urdu keywords instead of standard English-based syntax. Specific keywords are

targeted in this work, and these keywords are replaced with some other keywords taken from “Urdu”. These keywords are written in “English Typeface” i.e., the keyword “cin” is replaced with “keyBoardSay” whereas “screenKiTaraf” has taken the place of the “cout” keyword. Here new libraries will be written that will be used to translate partially written programs into pure C++ programs.

The core objective of this study is to reduce the linguistic barrier for novice programmers by enabling programming in native languages—specially Roman Urdu and Hindi—using a custom compiler extension named RUPT (Roman Urdu Programming Translator). RUPT serves as a preprocessor layer that translates Roman Urdu/Hindi code into syntactically valid C++ code, facilitating compilation through standard C++ compilers.

The significance of this problem lies in the widespread difficulty non-native English speakers face while learning programming, as most programming languages rely heavily on English-based syntax and semantics. This language-centric challenges often leads to cognitive overload, disengagement and slower learning curves.

The novelty of the proposed framework lies in its integration of native-language-inspired lexical tokens within a structured grammar system (RUPL), use of Finite State Machines (FSMs) for keyword recognition, and implementation of a custom tokenizer designed to handle non-standard script variations found in Roman Urdu. Key contributions of this work include:

- Designing and formalizing the RUPL grammar with mappings for Roman Urdu equivalents of common C++ keywords.
- Implementing a light-weight lexical analyzer that processes Roman Urdu-based source code and generates corresponding valid C++ code.
- Developing a fully functional GUI-based editor that supports writing and compiling programs in Roman Urdu.
- Conducting technical and opinion-based evaluations with students to validate ease of learning and system usability.

The rest of the article is organized in the following manner: In Section II, relevant work is elaborated. Section III presents the methodology used for state-of-the-art. Results and discussion, along with design and implementation, are covered in Section IV, while the conclusion is covered in Section V.

II. LITERATURE REVIEW

Over the last few years, several researchers carried out different studies for code conversion. David Unga et al. [9] proposed a computer-adaptive translator named the "University of Queensland Dynamic Binary Translator" (UQDBT) that followed a backward pass (decoding executable code) and forward pass (encoding the decoded executable code after required improvement). UQDBT converges faster than systems based on instruction anticipation because edge weight instrumentation converts frequently executed code to native code. It was a method that made it possible to run software on a machine and get acceptable output, whereas the software was designed for some other machine.

Tao Lei et al. [11] described a technique for creating input parsers automatically from English specifications of input file formats. The English specification was converted into a specification tree, which was then converted into a C++ input parser using a Bayesian generative model to capture pertinent natural language occurrences. A joint dependency parsing and semantic role labeling task was used to model the issue. Their approach is based on two different types of data: the first is the relationship between the text and the specification tree, and the second is noisy supervision, which is measured by how well the resulting C++ parser reads input examples. A state-of-the-art semantic parser obtained an F1-score of 66.7% using a dataset of input format specifications from the ACM (Association for Computing Machinery) International Collegiate Programming Contest, while this technique produced an F1-score of 80.0%.

Furqan et al. [18] released ERUPD, a parallel English-Roman Urdu corpus of 75,146 sentence pairs, created via synthetic prompt-engineering and human validation. This dataset could significantly enhance token mapping and keyword consistency in state-of-the-art related work. Besides, Ansarullah et al. [20] achieved 97.98% accuracy in segmenting mixed Roman Urdu and English text using dictionary based SVM and Bi-LSTM—highlighting effective strategies for script or code-switching detection application applicable to lexical analyzers.

RECCO, a REliable Code COmpiler that can automatically produce a reliable version of any C/C++ source code, was introduced by A. Benso et al. [12] The program uses a powerful algorithm for reordering the code and a flexible technique for variable duplication to build a trustworthy code that can recognize the appearance of important data defects. The tool makes changes that are entirely transparent to the programmer and have no impact on the targeted program’s original functionality. In order to keep overhead within the acceptable ranges, the tool also gives the user the option of selecting the percentage of duplicated variables. The approach’s efficacy and the low overhead that was added to the trustworthy code in terms of both memory occupancy and execution time were proved by experimental results.

Ben Gelman et al. [13] linked source code with three deliverables from 108,568 projects that were downloaded from GitHub and had at least 10 stars and a redistributable license. The first set of pairs links Doxygen-extracted comments with corresponding snippets of source code in C, C++, Java, and Python. The second group of pairings links the raw C and C++ source code repositories with the build artifacts that are

produced when the make command is used to create the code. The last set of pairs links unprocessed C and C++ source code repositories with probable code flaws, which are discovered by running the Infer static analyzer. The code and comment pairs can be utilized for tasks like comment prediction or code description in natural language. Reverse engineering and enhancing intermediate representations of code from decompiled binaries are two operations that can be accomplished using the code and build artifact pairs. It is possible to leverage the code and static analyzer pairs for machine learning approaches to vulnerability finding.

According to real-world applications and the drawbacks of existing programming software, Pan Duotao et al. [14] created the GIEPT, a type of cross-platform modelling programming software, on the open source Fedora 12/Linux platform. By utilizing its XML-based input programming approach and many programming kinds and scales, GIEPT offers a universal solution. The GIEPT now incorporates a number of features, such as the common solver interface, the unified standard using XML-based schema, which is used to input model data and automatically convert it into C++ source code, and the symbolic algebra system, which is used to produce the matrix of gradient, Jacobian, Hessian, etc. Since GIEPT is platform agnostic and open source, programmers are free to alter its characteristics and expand its functionality in response to the situation at hand. Additionally, development and application expenses have been significantly decreased.

In a C++ implementation of a concordance program for texts in Old West Norse and Runic Swedish, Lars Engebretsen [15] discussed some of the author's experiences. It was only reasonable to use Unicode to represent data both inside the program and in external files because the input to the program employed a character repertoire that no typical onebyte character encoding supports. The input and output were represented in UTF-8, while each character within the program was represented using C++ "wide characters." During file I/O, the author created C++ code conversion aspects that translate data between those two formats. This allowed him to successfully construct and execute the concordance application on both Windows XP (using Visual C++.NET 2003) and Linux (Fedora Core 3 with gcc 3.4.2). When switching platforms, only a few lines of code—the ones deciding which code conversion facet to use—had to be modified in the source code; all other sections of code stayed the same. Even though the code conversion facets given by the library had been updated, the author could still use the standard C++ locale framework for collation and code conversion.

A method for automatically creating documentation summaries for C++ procedures was suggested by Nahla J. Abid et al. [16]. A summary template was made using method stereotypes, one for each individual method archetype. The primary parts of the approach are then extracted using static analysis. The generated documentation summaries are then used to update each method's documentation. The strategy may be applied to various object-oriented programming languages and is very scalable. These summaries can aid in maintaining comprehension. Undergraduate students that participated in the evaluation were the initial subjects. The findings show that the automated summaries adequately describe what the approach accomplishes and contain all necessary details. The outcomes also suggest that their approach to this issue—creating unique templates for each stereotype—is a workable and effective remedy. Despite the fact that the automated summaries were generally praised by the participants, some changes are still required, notably for the controller and collaborator, because they are rather complex and challenging to effectively summarize.

III. METHODOLOGY

Butt et al. [17] propose a transformer-based model using m2m100 with masked language modeling to transliterate between Roman-Urdu and Urdu, achieving character-level BLEU scores of 96.37 and 97.44. Their use of transfer learning and rigorous domain adaptation offers a strong precedent for lexical mapping approaches in Roman Urdu Programming Translator (RUPT). RUPT will translate a program containing Urdu Roman words as an alternative to C++ keywords into a pure C++ program. It is able to perform lexical analysis of the program containing a mixture of Urdu Roman words and C++ keywords according to the definition of Roman Urdu Programming Language (RUPL).

A. RUPL DEFINITION

Roman Urdu Programming Language (RUPL) contains the following alphabets as C++:

{A, B, C, D, E, F, G, H, I, J, K,
L, M, N, O, P, Q, R, S, T, U, V,
W, X, Y, Z, a, b, c, d, e, f, g,
h, i, j, k, l, m, n, o, p, q, r,
s, t, u, v, w, x, y, z, 0, 1, 2,
3, 4, 5, 6, 7, 8, 9}

Real numbers can be defined by the following alphabets:

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, .}

TABLE 1: RUPL Keywords

C++ Keyword	RUPL Keyword	C++ Keyword	RUPL Keyword
cin	keyboardSay	cout	screenKiTaraf
if	agar	else	naheTw
for	klea	switch	badlo
case	imkan	break	roko
default	pehlySay	do	karo
exit	niklo	private	niji
public	awami	protected	mehfooz
while	jabK	return	wapis
continue	jari	string	doree

This table presents the mapping between C++ keywords and their equivalents in the proposed RUPL language.

The alphabets given above have been used to make grammar for our language, RUPL. Grammar has four

parts given below:

1) **N:** non-terminal alphabets
2) **T:** terminal alphabets
3) **P:** it defines production rules
4) **S:** it is start symbol belongs to N

Here is an example to define grammar for an identifier:

N = { <id>, <digit>, <letter>}
T = { 1 , 2 , 3 , a , b , c }
P = production rules to be used
1. <id> → <letter>
2. <id> → <id> <letter>
3. <id> → <id> <digit>
4. <digit> → 1
5. <digit> → 2
6. <digit> → 3
7. <letter> → a
8. <letter> → b
9. <letter> → c
S = <id>

B. RUPL KEYWORDS

Besides, a subset of C++ keywords has been selected. Among these, some keywords will be redefined into the Roman Urdu language. These Urdu Roman-redefined keywords will be replaced back into the original C++ by RUPT. Table 1 shows C++ keywords and their corresponding special RUPL keywords.

C. RUPL TOKENS

RUPT tokens are the fundamental structures that obstruct the Roman Urdu Programming Language, which are developed together to compose a RUPL program. Every single littlest individual unit in a RUPL program is known as a RUPL token. A few types of RUPL tokens given below:

1) Keywords, e.g., string, klea
2) Identifiers, e.g., total, main
3) Strings, e.g., school, university
4) Constants, e.g., 1001, 1136, 1089
5) Operators, e.g., *, -, +, /
6) Special symbols, e.g., { }, ( )

D. RUPL TRANSLATION

There are two essential qualities of simulated dialects: syntax and semantics. Language structure is an arrangement of standards that must be taken after to announce a legitimate program, while semantics depicts consistent conduct of the substantial program. The way toward contrasting source code and the punctuation of dialect is finished by the parser, while the code generator allocates implications to the program. Strategies used to determine the linguistic structure of any dialect are grammar, finite state machines, and regular expressions.

As discussed earlier, RUPL grammar has four parts: *N* , *T* , *P* , and *S*. An element from the non-terminal letter set, *N* , speaks to a gathering of characters from the terminal letters in order, *T* . A non-terminal image is as often as possible encased in edge sections, <>. While the guidelines of generation utilize the non-terminal to portray the structure of the language. Notice that *N* is a set, yet *S* is not. *S* is one of the components of set *N* . The beginning image, alongside the tenets of creation, *P* , empowers you to choose whether a series of terminals is a substantial sentence in the dialect. In the case of beginning from *S*, a series of terminals is created by utilizing the principles of generation; at that point the string is a legitimate sentence.

1) Grammar for Identifier

Despite the fact that a RUPL identifier can utilize any capitalized or lowercase letter or digit, the same as C++, to keep the case little, this punctuation allows just the letters l, m, and n and the digits 7, 8, and 9. The principal character must be a letter, and the rest of the characters, assuming any, can be letters or digits in any mix. This grammar has three non-terminals, namely, < *id* >, < *letter* >, and < *digit* >. The start symbol is < *id* >, one of the elements from the set of non-terminals. The rules of production are of the form: **A** −−· **w**, where *A* is a non-terminal and w is a string of terminals and non-terminals. The symbol −−· means “produces” while the grammar specifies the language by a process called a derivation. To derive a valid sentence in the language, begin with the start symbol and substitute for non-terminals from the rules of production until you get a string of terminals. Here is a derivation of the identifier nlm9 from this grammar. The symbol means −−· “derives in one stage”. Grammar for RUPL identifier is given below:

N = { <id>, <letter>, <digit>} T = { l , m , n , 7 , 8 , 9 }
P = shows rules of production
1. <id> → <letter>
2. <id> → <id> <letter>
3. <id> → <id> <digit>
4. <digit> → 7
5. <digit> → 8
6. <digit> → 9
7. <letter> → l
8. <letter> → m
9. <letter> → n S = <id>

Besides, each deduction step serves as the generation administrator upon which substitutions are based. For example, consider Rule 2:

$$\langle id\rangle \text{ --·} \langle id\rangle\langle letter\rangle$$

This rule is applied to substitute ⟨*id*⟩during the derivation stage. For instance:

$$\langle id\rangle 9 \text{ --·} \langle id\rangle\langle letter\rangle 9$$

The conclusion of this inference operation corresponds to performing substitution on a letter in sequence. The symbol
−−·* denotes "derives in zero or more steps". The last eight inference steps can be summarized as:

$$\langle id\rangle \text{ --·*} \text{ nlm9}$$

This derivation confirms that nlm9is a valid identifier, as it

can be generated from the start symbol ⟨*id*⟩.

2) Finite State Machine (FSM) to Parse Identifier

A 2024 study [21] on Roman Urdu spelling variation (5˜244 words per variant) emphasizes the prevalence of orthographic inconsistency—supporting RUPT's FSM-based normalization to correctly map variant tokens to standardized C++ keywords. In Figure 2(a), the arrangement of states {*A, B, C*} is given. *A* is the beginning state and *B* is the last state, whereas *C* is the reject state. There is progress from *A* to *B* on a letter, from *A* to *C* on a digit, from *B* to *B* on a letter or a digit, and from *C* to *C* on a letter or a digit. To utilize the FSM, envision that the information string is composed on a bit of paper tape. Begin in the beginning state, and output the characters on the information tape from left to right. Each time you examine the following character on the tape, influence a change to another condition of the limited state machine. Utilize just the change that is permitted by the curve relating to the character you have recently checked. Subsequent to filtering all the info characters, on the off chance that you are in a last express, the characters are a legitimate identifier. Else they are most certainly not. Figure 2(b) shows the same process through a simplified finite state machine by removing the optional reject state. Table 2 and 3 show transition tables for FSM Identifier and Identifier through Simplified FSM, respectively.

TABLE 2: Transition Table for Identifier

Current State	New State (Letter)	New State (Digit)
A B C	B B C	C B C

This table defines state transitions when processing identifiers: a letter leads to one transition, while a digit leads to another, based on the current state.

TABLE 3: Transition Table for Identifier through Simplified FSM

Current State	New State (Letter)	New State (Digit)
A B	B B	– B

This table shows a simplified finite state machine (FSM) used to identify valid identifiers. State transitions depend on whether a letter or digit is encountered.

By considering the following grammar, a few examples to parse RUPL keywords are represented.

X = { a, b, c, e, f, g, h, I, k, l, n, o, r, s, t, y }
T = { X, _ }
N = { <id>, <letter>, <symbol>} P = Production Rules
<id> → <id> <letter>
<id> → <id> <symbol>
<id> → <letter>
<letter> → X
<symbol> → _ S = <id>

Now, the key word "agar" will be parsed through the grammar defined above. As it is defined:

S = <id>
= <id><letter> (using Rule 1)
= <id><letter><letter> (using Rule 1)
= <id><letter><letter><letter>(using Rule 1)
= <letter><letter><letter><letter> (using Rule 3)
= agar

The keyword "agar" has been proved by RUPL grammar of keywords. Hence, "agar" belongs to RUPL. Figure 3 shows the FSM for the RUPL keyword "agar". The same as the keyword "klea" also being tested. So, the key word "klea" will be parsed through the grammar defined above. Parsing of the "klea" keyword is performed by applying the same rules as applied to parse "agar" because both keywords have 4 letters. Figure 4 shows FSM to parse RUPL keyword "klea".

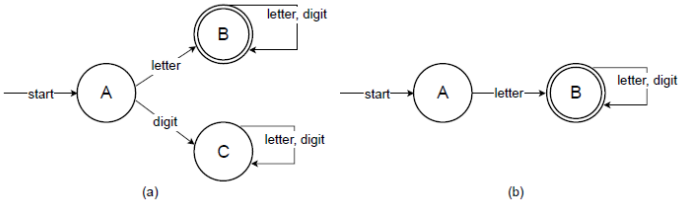

FIGURE 2: Finite State Machine (FSM) to Parse Identifier

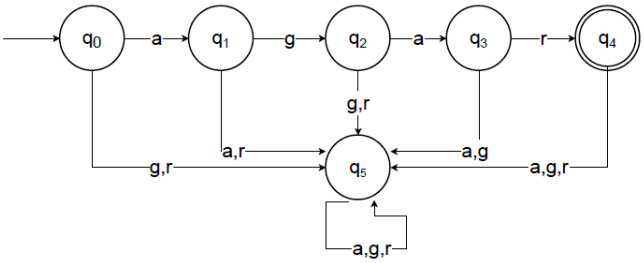

FIGURE 3: FSM to Parse RUPL keyword "agar"

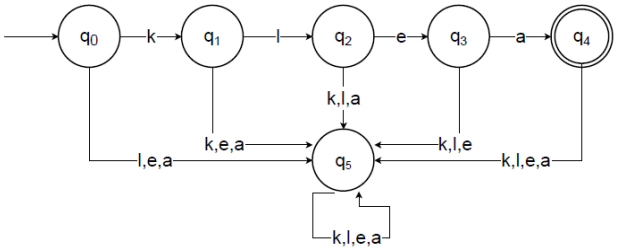

FIGURE 4: FSM to Parse RUPL keyword "klea"

The Algorithm 1 outlines how the RUPT lexical analyzer processes Roman Urdu keywords, identifieres, constants, and symbols to generate valid C++ tokens:

Algorithm 1 RUPT Lexical Analyzer with Spell Correction

Input: Source code written in RUPL (Roman Urdu Programming Language)
Output: Translated source code in C++ syntax
Load predefined `keywordMap` containing RUPL → C++ mappings
for all lines in RUPL source code **do**
  Tokenize each line into individual tokens $t_1, t_2, \ldots, t_n$
  for all token t_i **do**
    if $t_i \in$ keywordMap **then**
      Replace t_i with keywordMap[t_i]
    else
      Initialize $minDist \leftarrow 3$, $closest \leftarrow$ ""
      for all valid keyword k in keywordMap **do**
        $d \leftarrow \text{LevenshteinDistance}(t_i, k)$
        if $d < minDist$ **then**
          $minDist \leftarrow d$; $closest \leftarrow k$
        end if
      end for
      if $closest \neq$ "" **then**
        Replace $t_i \leftarrow$ keywordMap[$closest$]
        Display correction suggestion to user (optional)
      end if
    end if
    Append translated token to output
  end for
end for
Return the final translated C++ code

IV. RESULTS AND DISCUSSION

The benefit of the iterative model is that it facilitates the early development of a functional version of the product. As a result, implementing modifications is less expensive. This is the reason to follow the iterative development model for RUPT.

A. REQUIREMENTS

The requirement under which this framework is presented is to target those audiences facing problems in understanding the English language. It is known by everyone that the usage of compilers itself has all the keywords in the English language. That is the main reason RUPT is presented. To make the lives of those people easy by changing specific keywords. As a result, their programming skills won't be affected in any way. Other than that, during the process of the translation, natural language is also promoted, which results in better understandings and perceptions about the people that are doing this.

B. DESIGN

As everyone knows, a translator or compiler requires some keywords and a list of tokens by which it recognizes the source code and compares it with the grammar defined for the programming language and translates that programming language to another. Similarly, a list of keywords, presented in Table 1, has been designed and targeted by RUPT. The user writes RUPL source code in the RUPT editor and gets the required output as pure C++ code, and later on this code is passed to the traditional compiler for further processing. Figure 5 shows an abstract view of RUPT, while Figure 6 helps to present a detailed understanding of the state-of-the art translator.

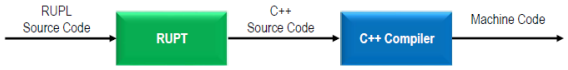

FIGURE 5: Abstract view of RUPT

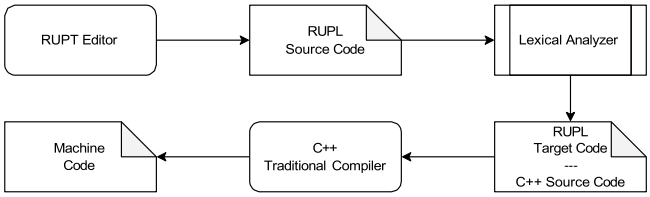

FIGURE 6: RUPT Detailed View

C. IMPLEMENTATION

After designing the RUPT, it was implemented into reality. For this purpose, C++ and C# have been used. DevCpp and Visual Studio are the main tools that have been used to implement the design. The RUPT editor has been developed as a user interface by writing instructions in C# as a programming language. As a tool, we have used Visual Studio to implement C#. Figure 7 shows the RUPT document window having RUPL source code and a RUPT interface. Source code to perform the process of converting a sequence of characters into a sequence of lexical tokens, called lexical analysis, is written in the C++ programming language. RUPT gives pure C++ code after performing lexing or tokenization on RUPL source code. Figure 8 elaborates on the working of the lexical analyzer.

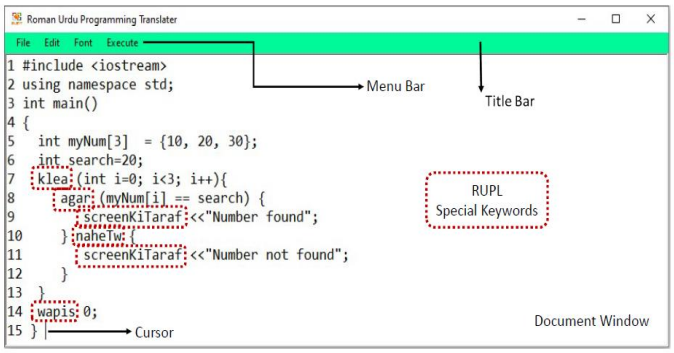

FIGURE 7: RUPT Interface

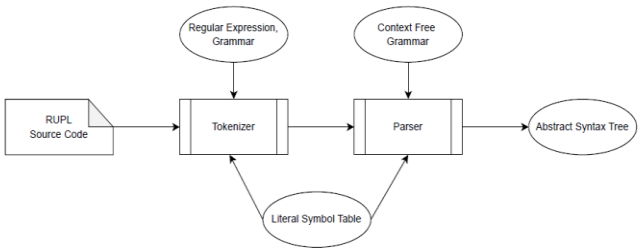

FIGURE 8: Lexical Analyzer Working

D. PROGRAM USED FOR EVALUATION

To verify the correctness of RUPT's translation output, a standard merge sort algorithm was used in Roman Urdu and then translated into C++ using the framework. The translated code was compiled and executed using a traditional C++ compiler. The Algorithm 2 shows the logic of the merge sort implementation used for evaluation:

Algorithm 2 Merge Sort Algorithm

```
Input: Array A, left index l, right index r
Output: Sorted array A
MERGESORT(A, l, r)
if l < r then
    m ← ⌊(l + r)/2⌋
    MERGESORT(A, l, m)
    MERGESORT(A, m + 1, r)
    MERGE(A, l, m, r)
end if

MERGE(A, l, m, r)
    n1 ← m − l + 1
    n2 ← r − m
    Create temporary arrays L[1 . . . n1] and R[1 . . . n2]
    for i = 1 to n1 do L[i] ← A[l + i − 1]
    for j = 1 to n2 do R[j] ← A[m + j]
    i ← 1, j ← 1, k ← l
    while i ≤ n1 and j ≤ n2 do
        if L[i] ≤ R[j] then A[k] ← L[i]; i ← i + 1
        else A[k] ← R[j]; j ← j + 1
        k ← k + 1
    end while
    while i ≤ n1 do A[k] ← L[i]; i ← i + 1; k ← k + 1
    while j ≤ n2 do A[k] ← R[j]; j ← j+1; k ← k+1
```

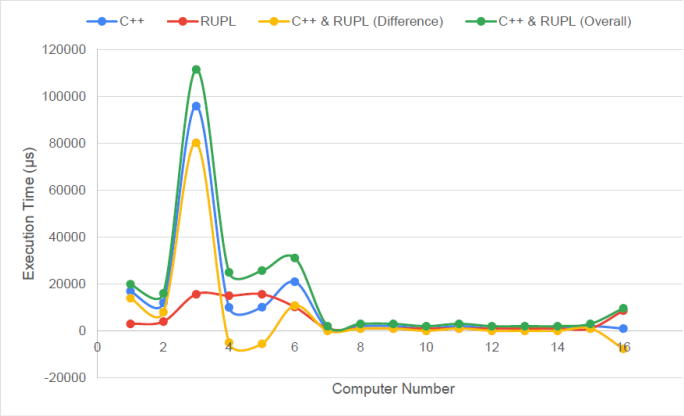

FIGURE 9: Sample View of Technical Servery

E. VERIFICATION

The community targeted to evaluate the state of the art was novel computer users towards programming, especially the intermediate students. The assessment process was based on two types of surveys. 1) Technical Survey and 2) Opinion Survey. In the technical survey, analysis was done by calculating the execution time of several programs, especially the merge sort program, by different users on many computers with variant aspects, e.g., CPU: Core, Frequency, and Generation. First of all, a pure C++ merge sort program was executed as aforesaid and calculated its execution time in microseconds. After that, a calculation was performed for the execution time taken by RUPT to yield a pure C++ program for merge sort from RUPL-based source code. A little bit of an increment in overall execution time was observed, but the understanding and learning rate increased as new learners found ease towards program scripting. A sample view of the technical servery is presented in Figure 9.

The opinion survey was conducted using both hard copy and digital forms, targeting intermediate-level and early undergraduate students from multiple educational institutions. Prior to participation, students were introduced to the use of RUPT in conjunction with RUPL for scripting, and its comparison with other programming languages. They were then asked to share their perspectives. The survey included students from both colleges and universities, representing a diverse academic background. The results based on responses from different institution types are illustrated in Figure 10.

In Figure 11, overall results present that 83.40% of participants appreciated RUPT as a great initiative to motivate individuals to the field of computer science, 8.30% said that in their views it does not affect them, while the same percentage remained neutral.

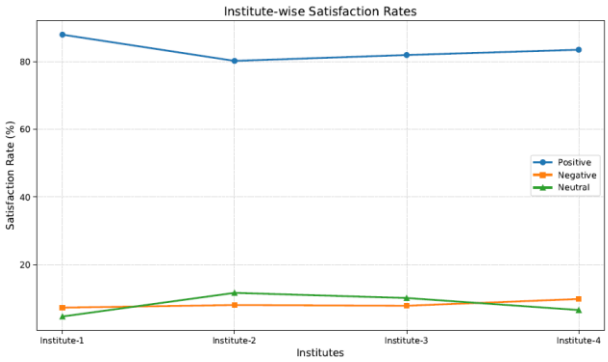

FIGURE 10: Views of Each Institute from Opinion Survey

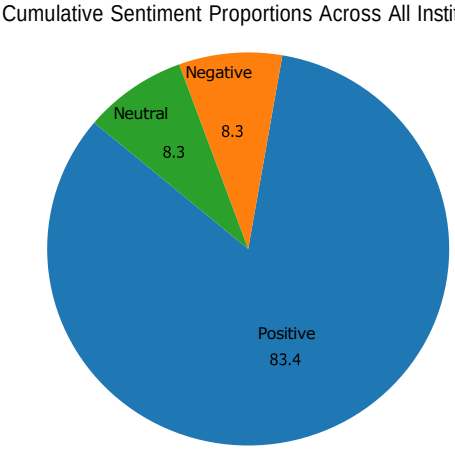

FIGURE 11: Overall Views from Opinion Survey

V. CONCLUSION

This work has proffered a framework, "Roman Urdu Programming Translator" (RUPT), as an additional layer to the original C++ compiler that translates a program coded in Roman Urdu or Hindi, known as "Roman Urdu Programming Language" (RUPL), into a proportionate C++ program. In this study, a special set of Roman Urdu keywords includes, e.g., “keyboardSay (ks)” instead of “cin”, “screenKiTaraf (skt)” instead of “cout”, “klea” instead of “for” etc., is focused on. In this work, additional Urdu and Hindi Roman keywords are added to the original set of C++ language and replaced by the C++ equivalent keywords through RUPT to convert it into a pure C++ program. RUPT is only composed of the lexical analysis phase. Keywords and tokens are defined and parsed by following the rules delineated in RUPL grammar. Evaluation is based on two types of surveys: 1) The technical survey has observed a minor increment in overall execution time, but the understanding and learning rate increased. 2) The opinion survey has presented that 83.40% of participants appreciated RUPT as a great initiative to motivate novel users to the field of computer science, 8.30% said that in their views it does not affect them, while the same number of participants remained neutral.

ACKNOWLEDGMENT

We would like to express our gratitude to the anonymous reviewers for their insightful feedback. This research work is not funded by anyone. Gramatical corrections are done by using AI at some places. Besides, authors do not have any conflict of interest.

REFERENCES

[1] J. Li, W. Cao, X. Dong, G. Li, X. Wang, P. Zhao, L. Liu, and X. Feng, “Compiler assisted Operator Template Library for DNN Accelerators,” *International Journal of Parallel Programming*, 2021, doi: 10.1007/s10766021-00701-6.

[2] N. P. Jouppi et al., “In Datacenter Performance Analysis of a Tensor Processing Unit,” *ISCA ’17*, pp. 1–12. Association for Computing Machinery, New York, NY, USA, 2017, doi: 10.1145/3079856.3080246.

[3] H. Liao, J. Tu, J. Xia, and X. Zhou, “DaVinci A Scalable Architecture for Neural Network Computing,” *2019 IEEE Hot Chips 31 Symposium (HCS)*, pp. 1–44. IEEE Computer Society, Los Alamitos, CA, USA, 2019, doi: 10.1109/HOTCHIPS.2019.8875654.

[4] V. I. Morosanova, I. N. Bondarenko, T. G. Fomina, and B. B. Velichkovsky, “Executive Functions and onscious Self-Regulation as Predictors of Native Language Learning Success in Russian Middle School Children,” *Journal of Siberian Federal University. Humanities & Social Sciences*, 2021, doi: 10.17516/1997-1370-0824.

[5] M. Daud, R. Khan, Mohibullah, and A. Daud, “Roman Urdu Opinion Mining System (RUOMiS),” *Computer Science & Engineering: An International Journal (CSEIJ)*, 2015, doi: 10.48550/arXiv.1501.01386.

[6] K. Mehmood, D. Essam, and K. Shafi, “Sentiment Analysis System for Roman Urdu,” *Proceedings of the 2018 Computing Conference*, 2018.

[7] A. Daud, W. Khan, and D. Che, “Urdu language processing, a survey,” *Artificial Intelligence Review, An International Science and Engineering Journal*, 2016, doi: 10.1007/s10462-016-9482-x.

[8] A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, “The IIT Bombay English-Hindi Parallel Corpus,” *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[9] D. Ung and C. Cifuentes, “Dynamic binary translation using runtime feedbacks,” *Science of Computer Programming*, 2005, doi: 10.1016/j.scico.2005.10.005.

[10] U. Hayat, A. Saeed, M. H. K. Vardag, M. F. Ullah, and N. Iqbal, “Roman Urdu Fake Reviews Detection Using Stacked LSTM Architecture,” *SN Computer Science*, 2022, doi: 10.1007/s42979-022-01385-6.

[11] T. Lei, F. Long, R. Barzilay, and M. Rinard, “From Natural Language Specifications to Program Input Parsers,” *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013.

[12] A. Benso, S. Chiusano, P. Prinetto, and L. Tagliaferri, “A C/C++ sourceto-source compiler for dependable applications,” *Proceeding International Conference on Dependable Systems and Networks. DSN 2000*, 2000, doi: 10.1109/ICDSN.2000.857517.

[13] B. Gelman, B. Obayomi, J. Moore, and D. Slater, “Source code analysis dataset,” *Data in brief*, 2019, doi: 10.1016/j.dib.2019.104712.

[14] P. Duotao, H. Mingzhong, and Y. Decheng, “Development of large scale programming system based on Linux platform,” *2011 Chinese Control and Decision Conference (CCDC)*, 2011, doi: 10.1109/CCDC.2011.5968310.

[15] L. Engebretsen, “Platform-independent code conversion within the C++ locale framework,” *Software — Practice and Experience*, 2006, doi: 10.1002/spe.734.

[16] N. J. Abid, N. Dragan, M. L. Collard, and J. I. Maletic, “Using stereotypes in the automatic generation of natural language summaries for C++ methods,” *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2015, doi: 10.1109/ICSM.2015.7332514.

[17] U. Butt, S. Veranasi, and G. Neumann, “Low-Resource Transliteration for Roman-Urdu and Urdu Using Transformer-Based Models,” *arXiv preprint arXiv:2503.21530*, 2025.

[18] M. Furqan, R. B. Khaja, and R. Habeeb, “ERUPD–English to Roman Urdu Parallel Dataset,” *arXiv preprint arXiv:2412.17562*, 2024.

[19] N. Hussain, A. Qasim, G. Mehak, O. Kolesnikova, A. Gelbukh, and G. Sidorov, “ORUD-Detect: A Comprehensive Approach to Offensive Language Detection in Roman Urdu Using Hybrid Machine Learning– Deep Learning Models with Embedding Techniques,” *Information*, vol. 16, no. 2, p. 139, 2025, doi: 10.3390/info16020139.

[20] S. H. Kumhar, M. Kirmani, S. Alshmrany, *et al.*, “Language Tagging, Annotation and Segmentation of Multilingual Roman Urdu-English Text,” 2024. [Online]. Available: https://arxiv.org/abs/ (DOI or publisher not available).

[21] M. A. Soomro, R. N. Memon, A. A. Chandio, M. Leghari, and M. H. Soomro, “A dataset of Roman Urdu text with spelling variations for sentence level sentiment analysis,” *Data in Brief*, vol. 57, p. 111170, 2024, doi: 10.1016/j.dib.2023.11117