International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Evaluating Intelligent Load Prediction Approaches for Reliable and FaultTolerant e-Governance Service Delivery

Sanjeev Kumar

Research Scholar, Department of Computer Science and Application, Om Sterling Global University Hisar, India

sanjeev.jangra.in@gmail.com, sanjeevcse221@osgu.ac.in

(Dr.) Saurabh Charaya,

Professor, School of Engineering and Technology, Om Sterling Global University, Hisar, India

Saurabh.Charaya@gmail.com

Dr. Rachna Mehta

Associate Professor, School of Engineering and Technology, Om Sterling Global University Hisar, India

drrachnamehta@osgu.ac.in

Abstract-. The fast growth of e-Governance services means that computing infrastructures need to be strong and efficient enough to handle changing workloads. To make sure that services keep running, resources are allocated ahead of time, and performance is optimized, it is important to be able to accurately estimate how much system resources will be used, especially CPU load. This research delineates the design and development of an innovative load prediction model for eGovernance services utilizing Artificial Intelligence (AI) methodologies, drawing on the Kaggle Borg Traces dataset, which comprises 405,894 entries documenting CPU, memory, and schedule metrics from Google's cluster management system. The methodology included a structured pipeline that included data pretreatment, feature extraction, and transformation. This included standardizing timestamps, parsing fields that seem like dictionaries or arrays, and dealing with missing values. Advanced feature engineering included low-variance filtering, removing highcorrelation features, making custom temporal features including lagged CPU utilization and rolling statistics, and applying Principal Component Analysis to reduce the number of dimensions. We trained and tested five regression models: XG-Boost, Gradient Boosting, Linear Regression, KNearest Next Door neighbors, and Support Vector Regressor. We used Mean Absolute Error (MAE), root average squared error (RMSE), and R^2 Score to do this. The findings indicate that XG-Boost surpassed all models, achieving MAE = 0.0001, RMSE = 0.0002, and $R^2 = 0.9998$, closely followed by Gradient Boosting with MAE = 0.0002 and $R^2 = 0.9992$, whilst SVM demonstrated inadequate generalization. The research demonstrates that AI-driven ensemble models are efficient for robust load forecasting, facilitating scalable, fault-tolerant, and citizencentric e-Governance frameworks.

Keyword-Load Prediction, Machine Learning, Services, CPU, Management and Variances.

I. INTRODUCTION

Governments throughout the world are quickly moving to digital systems, which has made eGovernance an important way to make service delivery more open, efficient, and focused on the needs of citizens. Governments are using integrated digital platforms more and more to handle important services including disaster response, healthcare, legal paperwork, public welfare programs, and taxes. These platforms often have to deal with huge amounts of requests that are hard to predict. These requests can change based on social and political events, legislative changes, seasonal needs, or even unexpected disasters. Because of this, the effectiveness and dependability of e-Governance structures depend on how well they can foresee, handle, and adapt to changes in workload needs in real time. In this instance, load prediction is highly crucial to make sure that the infrastructure's servers, databases, or communication networks perform smoothly without degrading the quality of service. Conventional load forecasting techniques are beneficial; nevertheless, they often struggle with the complexities or non-linear patterns seen in contemporary digital governance settings. This is particularly relevant when you think about how various users behave, how they get to things, and how outside social and economic variables affect them [1]– [3]. AI can help us get past these challenges by building prediction models that are more exact, flexible, and aware of their surroundings. AI-driven technologies, such as machine learning, deep learning, and hybrid models, have been very successful at finding complicated temporal patterns, linking multiple parameters, and developing accurate predictions that may be used to plan ahead and build systems that can handle errors. In the context of strong e-Governance, resilience signifies that the system can withstand abrupt spikes in demand, address performance issues, and swiftly return to normal after an interruption while not losing service quality. This requires a predictive engine that not only predicts load but also has ways to get input so that it can always improve its predictions and adapt to changes in how things work.

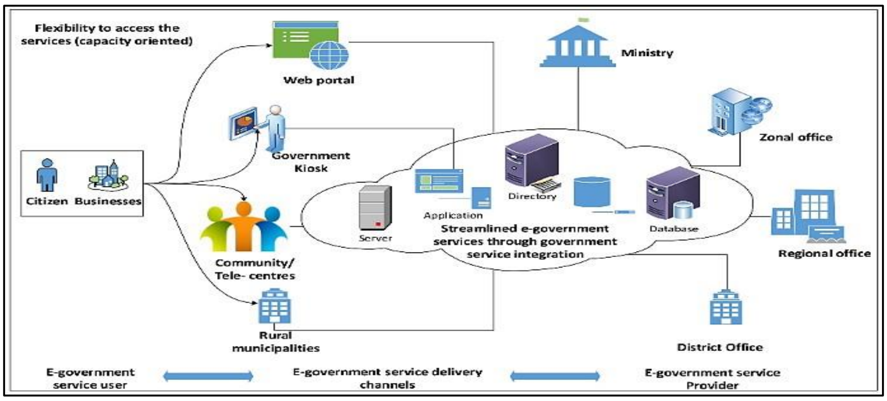
Flexibility to access the services (capacity oriented)
Web portal
Ministry
Zonal office
Government Kiosk
Citizen
Businesses
Directory
Application
Server
Database
Streamlined e-government services through government service integration
Regional office
Community/ Tele- centres
Rural municipalities
District Office
E-government service user
E-government service delivery channels
E-government service Provider

Fig. 1 Resilient e-Governance Services [4]

Therefore, a new AI-based workload prediction model must be developed with a multidimensional perspective that encompasses scalability for future growth, adaptability to evolving patterns, resilience to atypical occurrences, and compatibility with existing governance IT systems. Furthermore, this model must incorporate domain-specific components relevant to public digital services, like enforcement of policies cycles, citizen engagement trends, and compliance requirements for security and privacy [5], [6]. The development process entails the aggregation and preparation of extensive datasets from several sources, including online traffic logs, application usage metrics, network bandwidth utilization, and socio-economic factors. After then, advanced AI methods are utilized to make learning better over time. These methods include recurrent neural networks (RNNs), long short-term memory (LSTM) designs, and even attentionbased systems. Also, hybrid approaches that combine statistical approaches with AI models can deal with circumstances when there aren't many historical patterns or they are broken by new events. This makes forecasts more accurate. A load prediction model that works well makes eGovernance services more stable by letting resources automatically grow and shrink, enhancing load balancing strategies, and making it easier to plan for prepared incident response. In the big picture, it helps citizens trust the government more, make better use of infrastructure, and operate technology in a way that is beneficial for the environment [7], [8]. These are all positive things, however there are still certain flaws that need to be fixed when making a system like this. These include changes in data, the capacity to understand models for making policy selections, the ability to make predictions in real time, and the ability to work with safeguarding platforms to keep sensitive information safe [9]. This project aims to address these deficiencies by proposing and implementing a novel AI-driven load prediction model tailored for resilient e-Governance services. The proposed framework would combine deep learning techniques with ensemble methods to make predictions more accurate. It would also include strategies to continually learning and evolving. We will evaluate the proposed framework's efficacy versus conventional methods using actual e-Governance datasets. This will demonstrate that the proposed model excels in accuracy, durability, and operational efficiency [10]. This study aims to provide theoretical insights and feasible options for developing intelligent, future-ready management infrastructures that can deliver continuous, high-quality services under diverse operational conditions by reconciling advancements in AI with the specific requirements of e-Governance systems [11].

II. LITERATURE REVIEW

Matlala 2025 et al. examines the adoption of Virtual Evaluation (VE) in South Africa’s public sector, where digital transformation is increasingly prioritized. Despite growing interest, little empirical research addresses barriers to VE adoption in government M&E processes. Guided by the Technology Acceptance Model and Institutional Theory, the research uses workshops, surveys, and interviews to explore technological readiness, institutional support, and regulatory alignment. Key challenges include limited digital skills, inadequate infrastructure, unclear policies, and insufficient stakeholder engagement. Recommendations include capacity-building, infrastructure investment, and standardized policies to institutionalize VE, enhancing M&E effectiveness and supporting evidence-based governance through digital innovation [12].

Macabare 2025 et al. analyzes the proposed E-Governance Act of 2022, aimed at modernizing public services in the Philippines through digital technologies. The bill seeks to streamline transactions, enhance transparency, and expand service access via a unified online platform. It emphasizes system interoperability, cybersecurity, and digital inclusion. Using Quezon City as a case study, initiatives like the QC E-Services Portal and QCitizen ID System illustrate the bill's goals. Challenges such as outdated infrastructure, legal barriers, and limited digital literacy remain. highlights these issues and outlines essential measures for effective implementation, fostering more efficient, inclusive, and transparent governance [13].

Isah 2024 et al. investigates the impact of e-governance on service delivery at Nasarawa State University, Keffi (NSUK), focusing on online registration and school fee payment. Using a mixedmethods approach, data were collected from students, faculty, administrative staff, and IT personnel through surveys and questionnaires. Findings indicate that online registration has limited impact on service delivery, while online school fee payment is effectively implemented, improving efficiency and satisfaction. highlights the need for strategic planning, stakeholder engagement, and institutional readiness to maximize e-governance benefits, offering recommendations for enhancing public service delivery through ICT adoption in higher education institutions in Nigeria [14].

Chauhan 2024 et al. examines the essential components of resilient infrastructure for e-governance platforms, ensuring transparency, efficiency, and continuous service despite disruptions. It highlights the role of cloud computing, scalable architectures, disaster recovery, encryption, and cybersecurity frameworks in safeguarding governmental and citizen data. Emphasizing collaboration among IT teams, policymakers, and cybersecurity experts, the study analyzes global case studies to identify effective practices. It underscores the importance of regulatory compliance, including GDPR and India's Digital Personal Data Protection Act. With growing reliance on digital services, the paper presents a roadmap of technical, organizational, and policy strategies for secure, scalable, and reliable e-governance systems [15].

Maharjan 2024 et al. evaluates e-governance implementations in Nepal, focusing on quality assurance and technical aspects to improve service delivery. Recognizing that poorly implemented systems can worsen governance issues, it uses the E-taxation portal as a case study, assessing functionality, usability, reliability, portability, and performance. Findings highlight strengths and areas needing improvement, offering insights for enhancing system effectiveness. The research serves as a knowledge base for future e-governance projects, enabling government officials to reference best practices and address shortcomings. By improving implementation quality, Nepal can ensure its e-governance initiatives deliver efficient, reliable, and citizen-focused services, supporting better governance and public trust [16].

TABLE 1 LITERATURE SUMMARY

Authors/year	Methodology	Research gap	Findings

Fischer/2023 [17]	Legal analysis of data sovereignty.	Limited studies link data sovereignty with e-governance legal frameworks.	National laws significantly influence secure and effective egovernance implementation.
Sharmin/2023 [18]	Mixedmethods comparative case study.	Lack of comparative analysis on global e-governance adoption impacts.	E-governance boosts efficiency, transparency but faces adoption challenges.
Myeong/2023 [19]	Stakeholderinstitutional innovation interaction model.	Limited studies on innovation's moderating role in egovernance satisfaction.	Institutional and technological innovation enhance egovernance's stakeholder satisfaction impact.
Ajitha/2020 [20]	RNN-LSTM residential load prediction model.	Lack of accurate lockdown-specific residential load prediction models.	RNN-LSTM model accurately predicts residential load during COVID19 lockdowns.

III. RESEARCH METHODOLOGY

This study focuses on designing and developing a robust Load Prediction Model for distributed computing systems, utilizing the Kaggle Borg Traces dataset, which provides comprehensive information regarding resource usage and scheduling metrics from Google's Borg cluster management system. The methodology employed is meticulously organized, according to a systematic progression of data collecting, preprocessing, constructing features, visualization, and machine learning-driven predictive modeling. The workflow makes sure that estimating CPU utilization in large-scale computation clusters is repeatable, accurate, and easy to understand.

A. Data Collection

The main dataset used in this study is the Kaggle Borg Traces dataset, which can be found athttps://www.kaggle.com/datasets/ericgitonga/borg-traces. There are 405,894 records in this dataset from Google's Borg system, which has a lot of logs. The features show several aspects of how resources are used, such as how much CPU and memory are used, how jobs are scheduled, what priority levels are set, and how resources are actually used over time. This dataset is ideal for studying workload prediction, load balancing, and resource optimization because it is so thorough and contains a lot of time depth. There are category, quantitative, and even hierarchical dictionarylike structures in the dataset, therefore it needs to be pretreated and transformed before it can be modeled. The selected objective variable for this research is cpu_usage_mean, representing the average CPU usage during specified time intervals.

B. Data Preprocessing

It's hard to evaluate the raw Borg trace data dataset since it has a heterogeneous structure, entries that aren't always the same, and fields for numbers, categories, and objects. So, adequate preprocessing is highly crucial to make confident that the dataset is relevant for predictive modeling, full, and consistent. We painstakingly planned the preprocessing pipeline to fix these issues, and it had numerous processes occurring one after the other. The initial step was to work with the timestamps. This involved examining and modifying the raw time entries, which were stored in milliseconds. We detected and repaired errors like zeros or numbers that were too high as well as too low to keep calculations from being wrong or times from being inconsistent. After then, valid timestamps were changed into Python datetime objects. This made it easier to get timebased statistics like lagged CPU usage measurements, rolling averages, as well as other forms of temporal statistics. After standardizing the timestamps, a unique function was used to break out dictionary-like factors like resource-request, average_usage, or maximum_usage so that the CPU and memory parts could be taken out separately. Because of this improvement, it was feasible to make organized, unique features for each type of resource. This is necessary for good predictive modeling as information mining input. Fields that looked like arrays, such as cpu_usage_distribution and tail_cpu_usage_distribution, were changed from string arrays to numerical arrays. After that, descriptive statistics like the mean, maximum, minimum, and standard deviation were found. These data revealed both the average and the range of how resources were used, which gave a full picture of how work was done. Lastly, cleaning the data means getting rid of the original sophisticated object-type columns to save space and make the data easier to work with. The cluster column was then turned into a categorical variable to reduce space and make machine learning easier. All of these steps together generated a clean, organized, and high-quality dataset that is ideal for modeling work later on and makes sure that both time and resource use patterns are shown appropriately.

C. Exploratory Data Analysis (EDA)

Once preprocessing and feature modification were done, a full exploratory analysis of the data (EDA) was done to look for patterns, trends, connections, and oddities in the dataset. EDA is a critical step in figure out how resources are being used, choosing the proper features, and rendering machine learning models simple to understand. The first picture was a scatter plot that indicated how much memory and CPU were being used. This indicated groups, trends, and possible gaps in how nodes and processes were sharing the task. This first study let us find out if specific workloads were always utilizing too many or less system resources. Next, a feature correlation heatmap was made to check for multicollinearity among numerical attributes. This made it possible to find strongly correlated features and maybe filter them out to avoid redundancy and bias in predictive models. We used histograms and Kernel Density Estimation (KDE) to show the distribution of key resource metrics like CPU usage mean, maximum, provided memory, and requested memory. This helped us see skewness, outliers, and whole variability in resource use. Boxplots were also used to look at CPU utilization across different scheduling classes, showing how the medians and distributions of task priorities changed. Also, scatter plots that compared requested CPU usage to actual usage helped find tendencies of over-provisioning or underutilization, while violin plots showed how resource use varied amongst clusters. Line graphs of CPU standard deviation across time were used to look at temporal variations. These plots showed changes and stable workloads. These visual studies gave us important information about how data is spread out, how it changes over time, and any strange patterns that might be there. This information helped us come up with better feature engineering and selection tactics for building models.

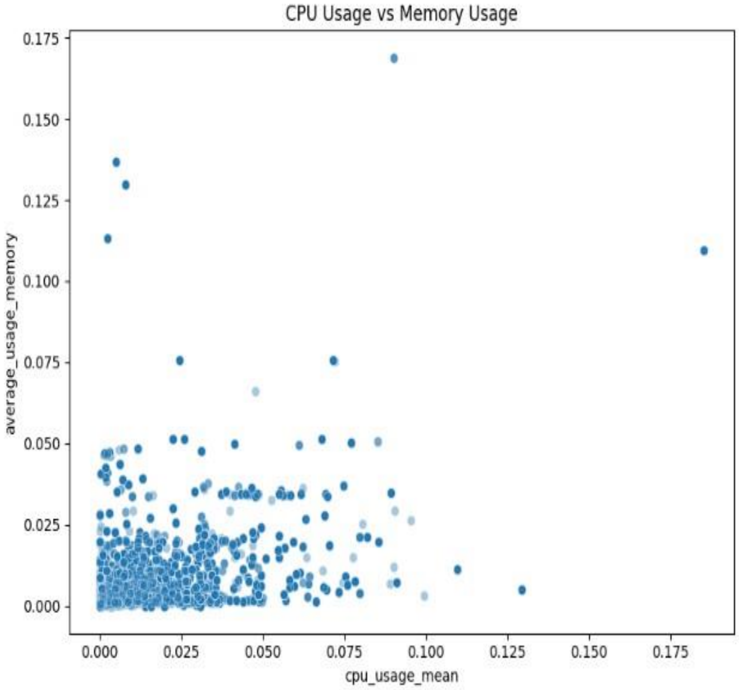

Fig. 2 Scatter Plot of CPU Usage vs Memory Usage

This scatter figure shows how the average CPU use and memory usage of all the processes in the B organ cluster are related. Each point stands for one process and shows how CPU usage and memory usage are related. Clusters of points show normal workload structures, while sparse or severe points show possible imbalances to resource bottlenecks. This visualization helps find jobs that use too many or too few resources, which helps with load balancing and allocation of resources in order to improve the overall performance of the system.

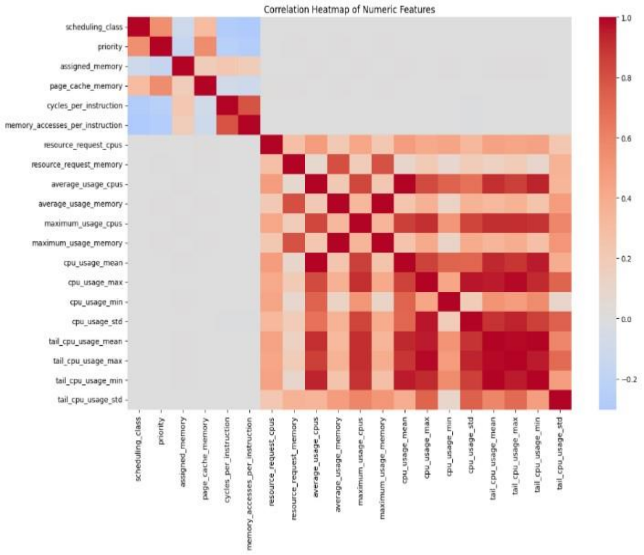

Fig. 3 Feature Correlation Heatmap

The heatmap shows how all the numerical attributes in the dataset are related to each other in pairs. The color intensity shows how strong the association is. Strong correlations can mean that there is redundancy to multicollinearity, which can hurt regression models if not fixed. Features that don't correlate well with others could give you unique prognostic information. This analysis helps choose the right features, get rid of unnecessary ones, and guide feature engineering. This makes sure that the last set of data used for modeling has the most predictive power while minimizing noise and the risk of overfitting.

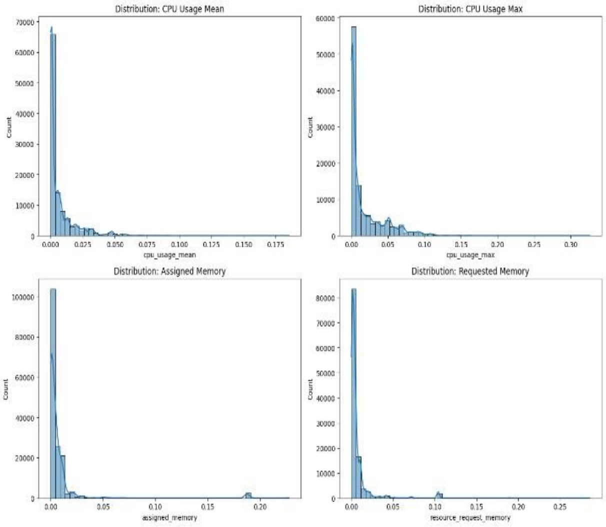

Fig. 4 Distribution of Usage Metrics

This picture shows histograms and Kernel Density estimator (KDE) curves for important resource consumption metrics, such as the mean and maximum CPU usage, the allotted memory, and the requested memory. It shows the distribution patterns, such as skewness, central tendency, and an abundance of outliers. By knowing these distributions, you may find problems, measure how much things change, and see whether there is a difference between what was requested and what was really used. This information is very important for preprocessing stages, making decisions about scaling, and making sure that the machine learning algorithm gets input data that is normalized and representative.

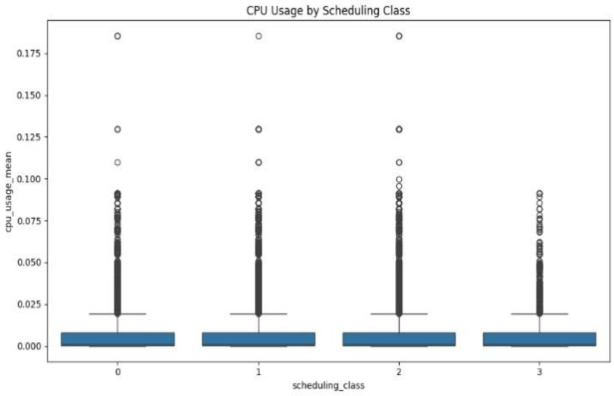

Fig. 5 CPU Usage across Scheduling Classes

The boxplot shows the mean CPU consumption for different scheduling classes, including medians, quartiles, and possible outliers for each task class. It shows clearly how the amount of CPU used changes based on the job's priority or kind. Classes with bigger interquartile ranges or severe outliers show that resource use is more variable. This helps with judgments about scheduling tasks, balancing loads, and allocating resources to specific tasks. It also helps find classes that routinely use too much or too little CPU resources, which can help with system optimization.

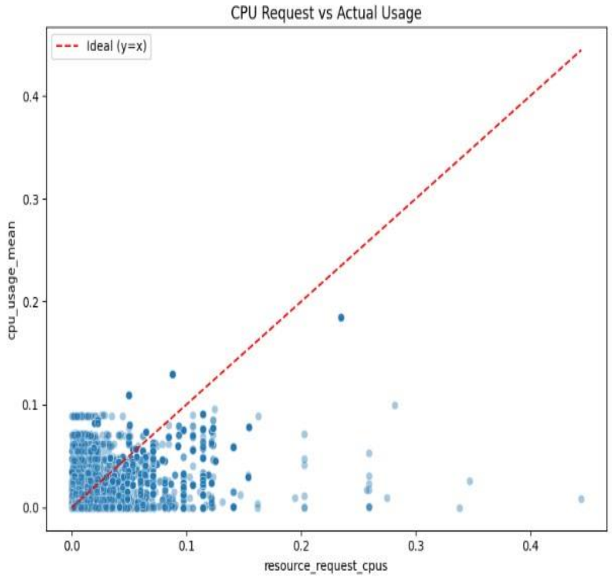

Fig. 6 CPU Request vs Actual Usage

This scatter plot shows how requested CPU resources relate to actual CPU usage. The ideal reference line (y = x) shows what perfect allocation would look like. If a point is above the line, it means that the request is too low. If it is below the line, it means that the request is too high or that the resources are not being used enough. The plot helps you find tasks that either use too much or too little CPU, which can help you improve your resource management tactics. By looking at how things differ from the standard, IT managers can improve resource allocation strategies, cut down on waste, and make the whole cluster work better.

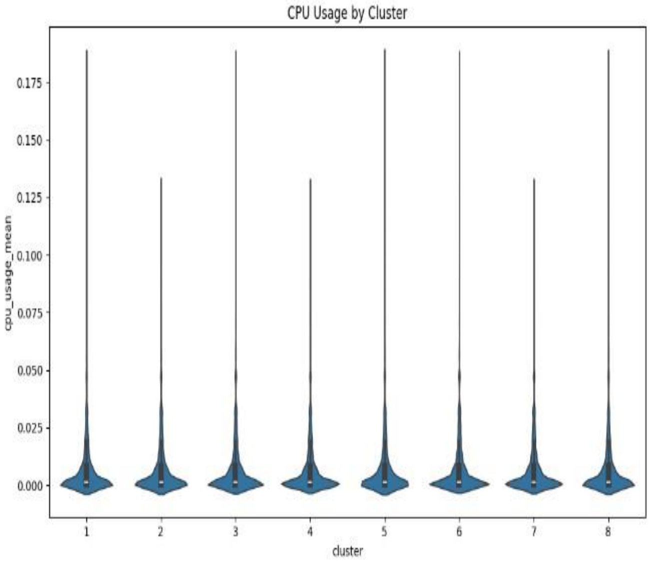

Fig. 7 Cluster-Level CPU Usage Differences

The violin plot shows how CPU utilization is spread out and how much it changes from one cluster to another. The width for each violin shows how many jobs there are at different degrees of CPU use, and the overall shape shows how workloads change amongst clusters. Wider distributions in clusters mean that CPU usage is more variable, while narrower violins mean that CPU usage is more stable. This graphic displays how smoothly the cluster is operating, points out probable problems or bottlenecks, and helps you figure out how to spread the load, make the cluster bigger, and determine the best ways to optimize it.

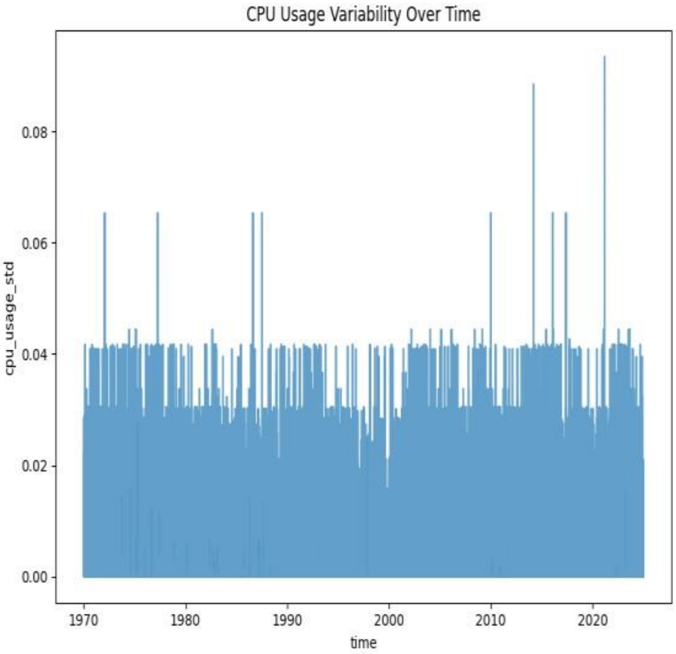

Fig. 8 Line plot of Variability in CPU Usage Over Time

This line plot shows how the average deviation of CPU usage evolves over time. It shows how stable the workload was over the time span that was measured. Peaks indicate periods of heightened performance stress or high variability, whereas troughs represent periods of constant resource utilization. We can learn more about how loads change over time, identify strange patterns, and make predictive models that consider into account how systems work in real time by looking at these changes over time. This picture helps with planning, giving out resources prior to time, and making changes to the system so that the cluster always runs at its best, even when workloads change.

D. Feature Selection

Feature selection is a crucial step to ensure that only pertinent and beneficial features are utilized in the creation of models. After preprocessing and exploratory data evaluation, rows with values that were lacking in the target variable processor use mean were removed to preserve the data clean for supervised learning. It was evident that the desired variable, cpu_usage_mean, was the average CPU usage per time period. The other columns were the attributes (X). We divided characteristics into two groups: qualitative variables, such as cluster identifiers, and numerical parameters, such as CPU and memory measurements. This split made it easier to encode, scale, and assess the data later on. Feature selection aims at two main goals: to make computations easier and to improve the model's efficiency by getting rid of extra or duplicate variables. To make predictive simulation work well, the dataset was carefully sorted and grouped by kind and how important each characteristic was to the goal. This made it less likely that there would be difficulties with noise and multicollinearity.

E. Feature Refinement and Transformation

After choosing the features, other methods were employed to clean up and change the data so it could be utilized to create better predictions. To get rid of low-variance traits that didn't add much information, we employed Variance Thresholding. This step made sure that only attributes with useful variability were maintained, which made the model operate better. Then, a correlation coefficient criterion of 0.9 was utilized to uncover traits that were quite similar to each other. To reduce multicollinearity, which could render regression models more inaccurate and harder to understand, redundant features were removed. We utilized ANOVA F-tests (f_regression) to see how important each number was to the aim variable cpu_usage_mean. This looked at how well each feature could guess what the target variable would be. This process created a list of features in order of their F-scores. This helped with both choosing features and made the model easier to grasp. We also employed feature scaling to make the range and spread of numerical information more even. Min-Max Scaling shifted the values to a range of [0,1], Z-Score standardization centered features close to zero with unit variance, along with Robust Scaling used the interquartile range as well as the median to make outliers less important. Final_data_minmax and final_data_zscore were two separate areas where scaled datasets were stored. This made it easy to make models of them later. This careful procedure of cleaning, normalizing, and optimizing the final dataset for machine learning algorithms made sure that it was ready to accurately estimate CPU usage.

F. Hand-Crafted Features and Dimensionality Reduction

Hand-crafted features were created to better predict changes in time and workload patterns. To describe temporal relationship, lag characteristics were incorporated, including CPU and resource request values from preceding intervals (lag 1 and lag 2). We employed rolling window indicators like the rolling median or standard deviation over three periods to discover short-term changes and trends. There was also a system status flag (high_cpu_flag) that showed when CPU usage was above the 80th percentile, which meant that resources were in high demand. We employed Principal Component Assessment (PCA) on standardized numerical traits to make things less complicated and uncover patterns that weren't obvious. This led to five main components that preserved most of the changes. The final set of features had the best initial features, time features that were designed, system flags, and PCA components. This made guaranteed that the input for training the model was both little and complete.

G. Dataset Splitting

To make sure the model was fully tested, the cleaned and feature-engineered dataset was divided into three parts: training, verification, and testing. The model could learn using just 70% of the data in the training set. We used the validation set, representing 10% of the data, to change the model's settings and keep it from overfitting. The remaining 20% was saved as a test set to check how well the predictions worked with new data. This splitting procedure made sure that the subsets had the same number of features and target values. It also made it easy to objectively assess performance, adjust hyperparameters, and get a credible estimate of how accurate predictions were.

H. Model Selection

This study selected five regression models to evaluate their efficacy in estimating CPU utilization from the refined Borg Traces dataset. We picked XG-Boost Regressor as the best choice since it has a gradient boost framework, can be made bigger or smaller, and can accurately deal with nonlinear interactions. We included support vector regression Regressor (SVR) to see how kernelbased methods perform with complex, data with high dimensions. Gradient Boosting Regressor used sequential learning of inadequate models to make estimates more accurate, which was an alternative technique to integrate models. We started with the linear regression approach because it was simple and easy to compare with more complicated models. Finally, the k-near-neighbor (KNN) Regressor was employed as a non-parametric method to detect patterns and similarities between instances and local workloads. These models included a wide range of machine learning approaches, such as linear, group- Based, kernel-driven, and distance-based. This made sure that all prediction methods for CPU load forecasting were thoroughly tested.

1. XG-Boost Regressor
2. Support Vector Regressor (SVR)
3. Gradient Boosting Regressor
4. Linear Regression
5. K-Nearest Neighbours (KNN) Regressor

I. Performance Metrics

We used three basic regression metrics—Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and a R² Score—to examine how well the models we projected functioned. MAE finds the average absolute difference between the actual and predicted values. This makes it easier to see how accurate a model is without being unduly influenced by major mistakes. RMSE, on the other hand, squares the terms that have to do with error, averages them, and then calculates the square root. This means that it is more responsive to bigger changes and gives a better idea of how well the model works as workloads change. The R² score (coefficient of choice) informs you how well the model explains the changes in the target variable. This is a nice way to see how well the model can make predictions. The evaluation utilized this combination of criteria to ensure that reliability, error sensitivity, and explanatory power were equitably assessed across all regression models implemented in the study.

IV. RESULTS AND DISCUSSION

The Results and Discussion section goes over all the different machine learning models and how well they operate at predicting CPU load in distributed computing settings. We used the improved but feature-engineered Borg Traces dataset to test the model's performance quantitatively using important metrics including Mean Absolute Error, the root-average square error (RMSE), and the coefficient of assessment. These measures look at how accurate the forecasts are, how sensitive they are to errors, and how well they explain things. There are also visual tests in this part that use real and expected scatter plots to show how well the model's forecasts match the actual CPU usage. This part compares new algorithms like XG-Boost or Gradient Boosting to older ones like KNearest Neighbors, lines regression, and Support Vector Regressor. It illustrates how well each model performs, how strong it is, and the way well it can be used in diverse situations. This lets us figure out how well each model can be utilized to guess the resilient load in online government services.

A. RMSE (Root Mean Square Error)

The RMSE is the square root of the standard deviation of the squared differences between the expected and actual values. It imposes higher penalties for bigger faults, which makes it valuable for seeing how well a model performs and how smoothly it can handle shifting workloads.

$$RSME = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}} \qquad (1)$$

B. Mean Absolute Error

MAE is a simple technique to measure how accurate an assumption is by showing the mean absolute variation between the predicted and actual values. It treats all mistakes the same, thus rendering it easier to comprehend but less likely to show major differences than RMSE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - x| \qquad (2)$$

C. Coefficient of correlation (R^2)

R^2 shows how much of the dependent variable's variance the model can explain. It measures total predictive strength, with values closer to 1 indicating high accuracy and negative values indicating poor fitting of models and generalization.

$$R^2 = \frac{n(\sum xy - (\sum x)(\sum y))}{\sqrt{[n\sum x^2 - (\sum x)^2]\,[n\sum y^2 - (\sum y)^2]}} \qquad (3)$$

Table 2 shows how well different machine learning models work by employing MAE, RMSE, and R^2 Score as metrics for analysis. The findings show that XG-Boost is the most accurate estimator, followed by gradient strengthening and Linear Regression. SVM, on the other hand, did very poorly, which means that it doesn't work well for predicting CPU utilization.

TABLE 2 PERFORMANCE EVALUATION OF ML MODELS

Model	**MAE**	**RMSE**	**R^2 Score**
XG-Boost	0.0001	0.0002	0.9998
SVM	0.0856	0.0864	- 49.3425
Gradient Boosting	0.0002	0.0003	0.9992
Linear Regression	0.0003	0.0005	0.9981
KNN	0.0004	0.0012	0.9895

The performance evaluation of the machine learning models in Table 2 shows that different methods have very diverse levels of predicted accuracy, as shown by the MAE, RMSE, and R^2 Score. The XG-Boost Regressor model performed the best of all the models. It had an MAE of 0.0001 and an RMSE of 0.0002, which were both very low. Its R^2 value of 0.9998 was practically perfect, showing that it was better at capturing complicated, non-linear connections in the dataset. The Gradient Boosting Regressor likewise did an amazing job, with an MAE of 0.0002, an RMSE of 0.0003, and a R^2 of 0.9992. It was quite similar to XG-Boost in terms of predictive power. Even though Linear Regression is a simpler baseline model, it got unexpectedly good results with $R^2 = 0.9981$. This shows that linear correlations still explain a lot of the variance in CPU consumption. K-Nearest Neighbors (KNN) did okay too, with a R^2 of 0.9895, although it wasn't as good as ensemble models. This is probably because it doesn't work well with high-dimensional data. The Support Vector Regressor (SVR), on the other hand, did not generalize well at all, with a negative

R^2 (–49.3425), which means it did not model the dataset well. This could be because it made wrong kernel assumptions or was too sensitive to scale. In general, ensemble methods like XG-Boost or Gradient Boosting did far better than traditional models. This makes them the best ways to predict CPU load.

 Actual vs Predicted Scatter Plot

To evaluate model performance visually, actual versus predicted scatter plots were generated for all five regression models. These plots provide an intuitive understanding of how closely model predictions align with observed CPU usage values. Ideally, data points should cluster tightly along the diagonal reference line (y = x), which represents perfect prediction accuracy. Figure 9 illustrates the performance of the Support Vector Regressor (SVR), where points are highly scattered away from the diagonal, confirming its poor predictive capability and negative R^2 value. In Figure 10, the K-Nearest Neighbours (KNN) Regressor demonstrates improved alignment, but slight deviations and dispersed clusters reflect its sensitivity to high-dimensional data and local workload variations. Figure 11 shows the Linear Regression model, where most points are closely aligned with the diagonal, demonstrating strong predictive ability despite being a simpler baseline model.

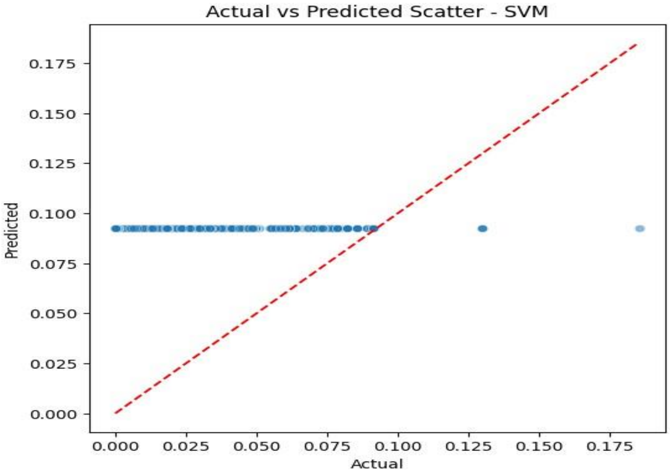

Fig. 9 Actual vs Predicted Scatter Plot-SVM

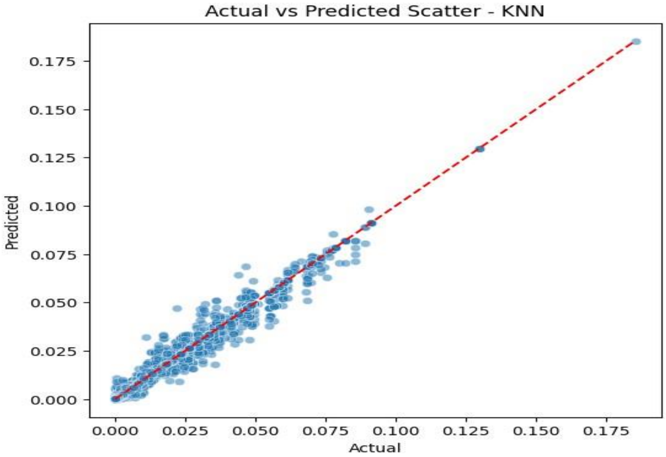

Fig. 10 Actual vs Predicted Scatter Plot-KNN

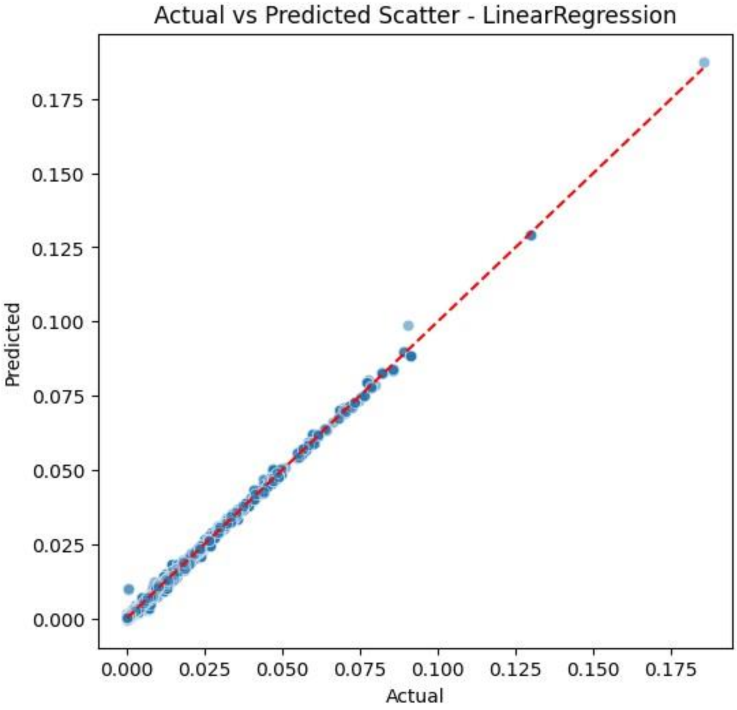

Fig. 11 Actual vs Predicted Scatter Plot-Linear Regression

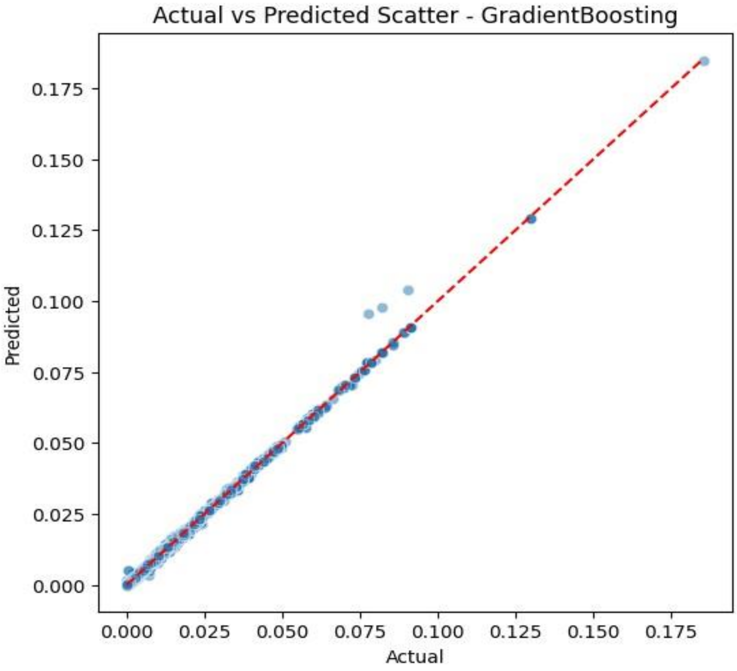

Fig. 12 Actual vs Predicted Scatter Plot-Gradient Boosting

Figure 12 highlights the Gradient Boosting Regressor, where data points almost perfectly match the reference line, confirming its robustness and high R² score. Finally, Figure 13 presents the XGBoost Regressor, which achieves the closest alignment with the diagonal, indicating near-perfect predictions and minimal error.

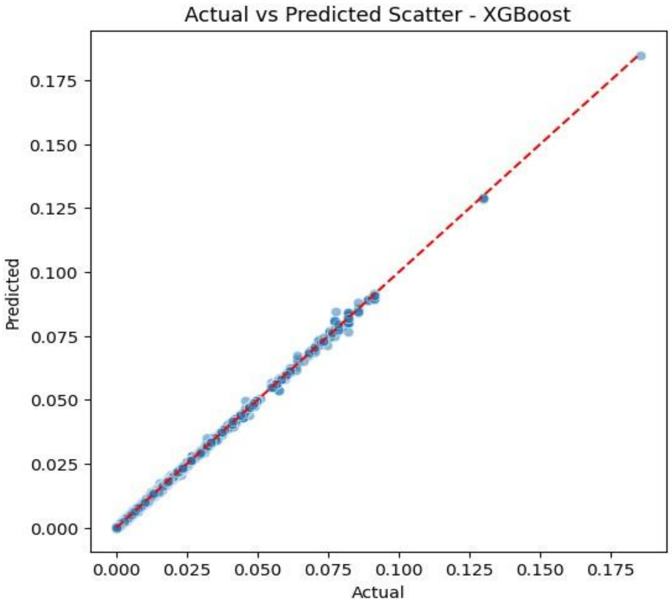

Fig. 13 Actual vs Predicted Scatter Plot-XGBoost

These visualizations confirm that ensemble models, particularly XG-Boost and Gradient Boosting, outperform others by capturing non-linear dependencies effectively. Meanwhile, Linear Regression offers competitive results, KNN shows moderate success, and SVR proves ineffective. Thus, scatter plot analysis reinforces the quantitative findings, establishing ensemble methods as the most reliable for CPU usage prediction.

V. CONCLUSION

This study presents the design and development of a novel load prediction model tailored for resilient e-Governance services using Artificial Intelligence (AI) techniques, with a special emphasis on CPU usage prediction in distributed computing systems. Using the Kaggle Borg Traces dataset comprising 405,894 entries and 34 attributes, the research applied structured preprocessing, feature engineering, exploratory data analysis, and multiple machine learning models to build a robust predictive framework. The preprocessing pipeline successfully transformed raw heterogeneous data into clean and structured features, while advanced techniques such as correlation filtering, variance thresholding, handcrafted temporal features, and PCA-based dimensionality reduction enhanced model efficiency. Performance evaluation using MAE, RMSE, and R^2 Score revealed that ensemble-based models achieved the highest accuracy. XG-Boost emerged as the best-performing model with MAE = 0.0001, RMSE = 0.0002, and $R^2 = 0.9998$, demonstrating near-perfect predictive strength. Gradient Boosting followed closely with MAE = 0.0002, RMSE = 0.0003, and $R^2 = 0.9992$, while Linear Regression also delivered competitive accuracy ($R^2 = 0.9981$). KNN achieved moderate success with $R^2 = 0.9895$ but exhibited higher RMSE (0.0012), reflecting sensitivity to high-dimensional data. Conversely, SVM significantly underperformed, yielding MAE = 0.0856, RMSE = 0.0864, and a negative R^2 of –49.34, confirming poor generalization. Visual validation through actual versus predicted scatter plots further reinforced these findings, where XG-Boost and

Gradient Boosting demonstrated nearperfect alignment with the diagonal line of ideal predictions. Overall, the results confirm that ensemble models, particularly XG-Boost, provide the most resilient and scalable predictive performance, making them highly suitable for dynamic workload management in e-Governance systems. By enabling accurate CPU load forecasting and proactive resource allocation, the proposed model enhances system resilience, scalability, and service continuity, ensuring robust, fault-tolerant, and citizen-centric digital governance infrastructures capable of meeting the evolving demands of large-scale service delivery.

REFERENCES

[1] K. M. A. Aziz, A. O. Daoud, A. K. Singh, and M. Alhusban, "Integrating digital mapping technologies in urban development: Advancing sustainable and resilient infrastructure for SDG 9 achievement – a systematic review," *Alexandria Eng. J.*, vol. 116, no. January, pp. 512–524, 2025, doi: 10.1016/j.aej.2024.12.078.

[2] M. D. Sajorda, M. V. Abante, and F. Vigonte, "Best Practices In E-Governance -Civil Service Commission(CSC)," *SSRN Electron. J.*, 2025, doi: 10.2139/ssrn.5279700.

[3] T. Ravago, M. V. Abante, and F. Vigonte, "<p>Digitalizing Justice: A Narrative Review of E-Governance Initiatives in the Philippine Department of Justice</p>," *SSRN Electron. J.*, 2025, doi: 10.2139/ssrn.5276721.

[4] "E-Government Maturity Model for Sustainable E-Government Services from the Perspective of Developing Countries."

[5] R. Mahant, "Artificial Intelligence in Public Administration: A Disruptive Force for Efficient E-Governance," *Mach. Intell. Res.*, no. January, 2025.

[6] A. Hardy, "Estonia's digital diplomacy: Nordic interoperability and the challenges of crossborder e-governance," *Internet Policy Rev.*, vol. 13, no. 3, pp. 0–31, 2024, doi: 10.14763/2024.3.1785.

[7] S. Lubis, E. P. Purnomo, J. Ahmad, and C.-F. Hung, "E-Governance and Sustainable Development Goals: A Systematic Literature Review," *Res. Sq.*, pp. 1–16, 2024.

[8] A. L. Maureal, M. A. E. Telen, U. L. F. Uy, and F. M. A. Lorilla, "Enhancing e-Governance through Microservices - The Development and Impact of the NTC-EDGE System," *Mindanao J. Sci. Technol.*, vol. 22, pp. 260–276, 2024, doi: 10.61310/mjst. v22iS1.2221.

[9] U. Bullanday, "Best Practices in E-Governance of the Commission on Elections: A Narrative Review of Objectives, Pillars, Best Practices, and Persistent Challenges with a Case Study of the 2016 Philippine National Election.," *Pillars, Best Pract. Persistent Challenges with a Case Study 2016 Philipp. Natl. Elect.*, vol. 2, no. 1, pp. 1–12, 2025.

[10] E. K. Isabirye, "Securing E-Governance Communication between Nations, States: A Cross- Border," no. July, 2025.

[11] N. Morze, R. Makhachashvili, V. Zvonar, L. Ilich, and M. Boiko, "Navigating The Digital Frontier: Ukraine's E-Governance Curriculum Amidst Crisis and EU

Integration," *Proc. World Multi-Conference Syst. Cybern. Informatics, WMSCI*, vol. 2024-September, no. Wmsci, pp. 4–9, 2024, doi: 10.54808/WMSCI2024.01.4.

[12] L. S. Matlala, "E-governance in South Africa: barriers and enablers of virtual evaluation in the public sector," *Insights into Reg. Dev.*, vol. 7, no. 2, pp. 84–108, 2025, doi: 10.70132/d9854558432.

[13] A. G. Macabare, M. V Abante, and F. Vigonte, "The Road to Digital Government: Exploring the E-Governance Act of 2022 and Quezon City's Local E-Government Initiatives," 2025.

[14] I. S. Isah, A. A. Chiroma, and A. M. Dance, "Assessment of E-Governance Implementation on Service Delivery in Nasarawa State University, Keffi (2017-2021)," *AKSU J. Adm. Corp. Gov.*, vol. 1, no. 1, pp. 89–100, 2024, doi: 10.61090/aksujacog.2024.041.

[15] A. Chauhan, "Designing Robust and Scalable Infrastructure Solutions to Ensure High Availability and Security in E- Governance Platforms," vol. 2, no. 6, pp. 1–4, 2024.

[16] S. Maharjan, P. De Chang, and D. Shrestha, "Evaluation on taxation portal of E-Governance system in Nepal.," *Researchgate.Net*, no. November 2019, 2024.

[17] A. Fischer, "Data Sovereignty and E-Governance: The Legal Implications of National Laws on Digital Government Systems," *Leg. Stud. Digit. Age*, vol. 2, no. 4, pp. 1–12, 2023.

[18] S. Sharmin and R. H. Chowdhury, "Digital transformation in governance: The impact of egovernance on public administration and transparency," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 1, pp. 362–379, 2023, doi: 10.32996/jcsts.

[19] S. Myeong and S. A. A. Bokhari, "Building Participative E-Governance in Smart Cities: Moderating Role of Institutional and Technological Innovation," *Sustain.*, vol. 15, no. 20, pp. 1–23, 2023, doi: 10.3390/su152015075.

[20] "Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19. The COVID19 resource centre is hosted on Elsevier Connect, the company' s public news and information," no. January, 2020.