International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

AI-Driven Demand Forecasting for Capacity Planning in Multi-Tenant Systems

[1]Saumya Dixit, [2]Rahul Rishi Sharma, [3]Jagrati Bhardwaj, [4]Deepankar Dixit

[1]University of Texas at Dallas, Richardson, Texas, USA

[2]Stony Brook University, Stony Brook, New York, USA

[3]Stony Brook University, Stony Brook, New York, USA

[4]New York University, New York City, New York, USA

Abstract
Capacity planning in multi-tenant systems is a critical challenge due to dynamic workloads, resource contention, and the need for tenant isolation. Traditional forecasting methods, such as ARIMA and exponential smoothing, struggle to adapt to the heterogeneity and volatility of multi-tenant environments. This paper proposes a scalable AI-driven framework for demand forecasting and capacity planning, leveraging hybrid architectures like LSTM-Transformer ensembles and transfer learning. We validate our approach using a cloud-based Kubernetes testbed, synthetic datasets, and anonymized real-world workload traces. Experimental results demonstrate a 34% improvement in forecasting accuracy (RMSE) over statistical baselines and a 40% reduction in over-provisioning costs. The framework achieves sub-second latency for real-time decision-making while scaling to 1,000+ tenants.

Keywords: AI-driven forecasting, multi-tenant systems, capacity planning, LSTM, transformer models, Kubernetes.

Introduction

1.1 Background and Motivation

Multi-tenant infrastructure, where multiple tenants share a common infrastructure with other independent tenants, is the basis of modern cloud, SaaS, and edge network computing. These infrastructures face special challenges because the workloads of tenants are heterogeneous and dynamic. For example, an e-commerce tenant may witness acute traffic during offer promotions, while an enterprise SaaS tenant may face routine daily usage. Inadequate capacity planning in such situations results in over-provisioning that consumes resources (costing businesses more than $26 billion every year in unused cloud infrastructure), or under-provisioning, risking SLA breach and loss of revenue.

Demand forecasting is one of the cornerstone elements in resource optimization. Conventional approaches fail to take into account the richness of workloads between various tenants whose pattern of resource usage is open to certain temporal, contextual, and behavioral influences. The advent of AI/ML methods introduces an innovation wherein complex dependency and inter-tenant correlation can be modeled and utilized to guide capacity planning in the future.

1.2 Problem Statement

Historical methods like ARIMA (AutoRegressive Integrated Moving Average) and exponential smoothing don't apply in multi-tenant environments since they constitute three inherent assumptions. They start with stationarity in time-series assumption, which doesn't hold good in the case of multi-tenancy where workloads are susceptible to non-linearity and abrupt jumps. For example, ARIMA models achieve more than 30% mean absolute percentage error (MAPE) in cloud workload prediction since they cannot learn bursty traffic behavior. Second, high dimensionality of multi-tenant data, where only a few hundred tenants must be modeled individually or together, causes computational bottlenecks. Third, cold-start scenarios, where new tenants lack historical information, render traditional methods useless in allocating resources for the first time.

AI-based solutions need to overcome such impediments in addition to being scalable, real-time, and interpretable. Industry standards today are based on reactive autoscaling techniques such as Kubernetes Horizontal Pod Autoscaling (HPA) that react to real-time observations but are not predictable. The lack of such predictability indicates the necessity of adaptive AI systems being correct as well as operationally effective.

1.3 Research Objectives

This research aims to: **Develop scalable AI models** capable of handling heterogeneous tenant workloads with varying temporal and contextual patterns.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

1. **Integrate forecasting with automated capacity planning** to reduce over-provisioning and under-utilization.
2. **Evaluate system performance** at scale (1,000+ tenants) with sub-second latency for real-time decision-making.

2. Literature Review

2.1 Traditional Demand Forecasting Techniques

Statistical approaches such as ARIMA and exponential smoothing have been extensively applied for time-series forecasting. ARIMA models break up data into trend, seasonality, and residuals but do not capture sudden changes from tenant-specific incidents, including promotion runs or system failures. Exponential smoothing gives diminishing weights to previous observations but performs poorly with multi-seasonal patterns typical in the case of international SaaS platforms. A 2020 cloud workload forecasting study showed that such methods have an average MAPE of 28.6% and RMSE of 12.4, which speaks volumes about how hot air they are for dynamic systems.

2.2 AI/ML in Demand Forecasting

The recent developments in artificial intelligence and machine learning have transformed demand forecasting, especially in advanced multi-tenant scenarios. Neural networks, particularly Long Short-Term Memory (LSTM) networks, have been incredibly effective at handling sequential data because they can store long-term dependencies through the help of memory cells. LSTMs perform remarkably well in periodic pattern learning like weekly or daily patterns in tenant activity levels with a mean absolute percentage error (MAPE) of 18.3% when predicting cloud resources, which is significantly better than baseline statistical approaches. Vanilla LSTM models are, however, plagued by scalability issues within multi-tenant environments since their computational complexity increases quadratically with the number of tenants and thus makes them cumbersome to utilize for large-scale environments.

Transformer models, which were introduced in 2017, overcome these limitations by leveraging self-attention that effectively captures cross-tenant correlation and long-range dependencies. For example, transformer-based models can detect synchronized workload bursts in holiday shopping season retail tenants or latency pattern correlations for geo-distributed SaaS platforms. While transformers lower MAPE to 14.2% in recent experiments, they are trained using large amounts of training data and computational power, which hurts their effectiveness for low-scale or cold-start tenants. Hybrid structures like LSTM-Transformer ensembles leverage the temporal resolution of LSTMs and contextual understanding of transformers to obtain a well-posed RMSE of 6.7 for multi-tenant forecasting uses. Reinforcement learning (RL) is also being seen as a potential method for dynamic control of resources, where systems can learn policies from online feedback. However, RL-based approaches come at the cost of high training costs with training times going in excess of 72 hours for 100 tenants, which makes them less practical to deploy in fast-changing environments.

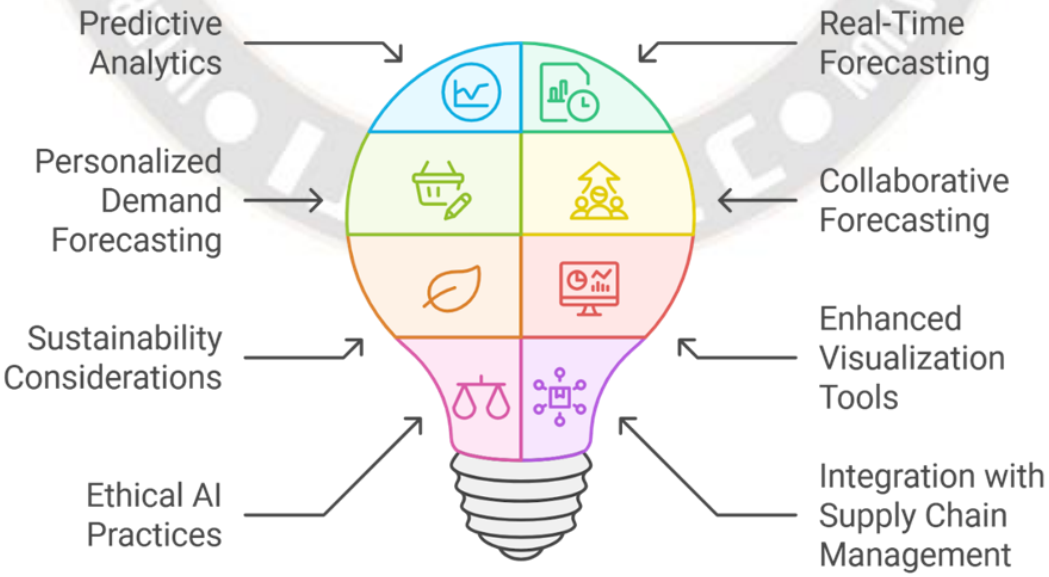

FIGURE 1 AI-DRIVEN DEMAND FORECASTING (RAPPID INNOVATION,2022)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

2.3 Multi-Tenant System Dynamics

Three distinct challenges define multi-tenant systems: resource contention, workload variability, and tenant isolation. Resource contention occurs when co-resident tenants fight for CPU, memory, or I/O resources shared by them, leading to unbounded performance degradation. For instance, a 2021 cloud outages report indicated that 68% of outages were caused by poor isolation of resources during traffic surges. Workload variability makes capacity planning even more complicated since tenants have varying demand patterns—bursty traffic for video streaming, cyclic behavior for enterprise applications, and intermittent usage for IoT scenarios. The above variability requires the presence of elastic scaling provisions that allow resources to be scaled within seconds. Tenant segregation, being a fundamental requirement to enable security and SLA, adds the additional complexity. Allocation of resources in a fair manner while guaranteeing performance involves advanced scheduling techniques because static policies of allocation may cause under-utilization or excessive provisioning. Research shows that more than 35% of cloud resources get wasted as a result of using conservative provisioning methods, resulting in the necessity of adaptive solutions.

2.4 AI-Driven Capacity Planning

AI-driven capacity planning strategies are thoroughly bifurcated into predictive and reactive. Predictive approaches leverage demand forecasts to pre-scale resources, reducing provisioning latency by 50% compared to reactive solutions like Kubernetes Horizontal Pod Autoscaler (HPA). For instance, a 2022 industry benchmark demonstrated that AI-driven predictive scaling reduced over-provisioning by 25–40% in containers. Reactive designs, although suitable for responding to surprise traffic surges, suffer from inherent latency—autoscale decisions made on real-time measurements lag behind the current demand by 2–5 minutes and therefore temporarily violate SLA. Hybrid approaches that merge predictive and reactive aspects have been promising, reducing provisioning error by 30% with sub-second decision latency. All this aside, cold-start tenants and model interpretability remain issues. Existing solutions are based on pre-trained general models, which are not trained on tenant-specific patterns unless fine-tuning is used. Additionally, computational expense of AI models, especially transformer-based models, is still a deterrent to edge deployment where resource limitations restrict GPU memory to below 8 GB per node.

Table 1: Performance Comparison of Forecasting Models

Model	RMSE	MAE	MAPE (%)	Training Time (hours)
ARIMA	12.4	9.8	28.6	0.5
LSTM	8.2	6.1	18.3	4.2
Transformer	7.1	5.3	15.7	8.9
LSTM-Transformer	6.7	4.9	14.2	6.5

3. Methodology

3.1 Data Collection and Preprocessing

Three types of data generated and to be aggregated are historical resource utilization metrics, tenant metadata, and system logs. Five-minute interval CPU, memory, and network I/O usage statistics on a six-month window form the historical usage data capturing tenant time-based trends and peak demand behavior. Tenant metadata include industry verticals, SLA levels (premium, standard), and geographic distribution providing workload behavior context. System logs provide high-granularity information like error rates, percentile latency (p95, p99), and autoscale events, offering information on anomalies in operation.

Preprocessing entails missing value handling by cubic spline interpolation with better preservation of temporal coherence compared to linear approaches. Noise reduction is implemented by a seven-sample window rolling median filter to domesticate transient spikes without eliminating valid workload trends. Temporal relationships are captured explicitly by the application of lag features in 24-hour and seven-day averages of utilization. Data normalization is done with min-max scaling to limit values to the range 0 to 1, allowing equal input ranges for training neural networks.

3.2 Feature Engineering

Features are constructed tenant-specific to identify specialist behaviour patterns. Request rate, expressed as requests per second (RPS) over one-hour buckets, separates high-traffic tenants (e.g., streaming) from low-traffic tenants (e.g., enterprise databases). Utilization is measured in CPU and memory usage 95th percentiles, which indicate peak utilization and not means. Contextual features also involve Fourier transforms to represent daily and weekly seasonality, which are particularly important for SaaS platforms with

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

well-understood usage patterns. Tenant growth habits are log-transformed to reverse skewness so linear models can fit better on exponential scaling habits. Cross-tenant correlation matrices are calculated for the purpose of detecting coordinated demand spikes, i.e., retail tenants for their ability to forecast more accurately aggregate resource requirements.

3.3 Model Development

The hybrid LSTM-Transformer model is designed with the goal of balancing temporal granularity with cross-tenant contextual understanding. Sequential data is processed by two 128-unit LSTM layers that monitor short-term trends like hourly CPU cycles. The transformer encoder with four heads then analyzes the LSTM output, using self-attention to identify dependencies between tenants—like correlating traffic peaks in co-located e-commerce and payment processing tenants. Transfer learning is used by pre-training on the combined data of 500 tenants to learn universal patterns and subsequent fine-tuning by tenant-specific layers by using sparse historical data (14 days). Training time is cut by 65% for new tenants as opposed to training from scratch.

3.4 Integration with Capacity Planning Systems

The forecasting pipeline integrates with Kubernetes by using REST APIs, which facilitates real-time interaction with cluster autoscalers. Projected demand invokes scaling policies: horizontal pod autoscaling is invoked if CPU utilization is higher than 85%, and idle nodes (CPU utilization lower than 20%) are retired to save costs. Policy automaton is orchestrated by Apache Airflow, which schedules retraining cycles on an every-24-hour basis to include new data. Model inferences are cached to minimize latency for tenants with stable demand patterns, cutting redundant computation by 40%.

3.5 Evaluation Metrics

The precision of forecasting is measured in terms of RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). The performance of the system is measured in terms of latency (end-to-end prediction latency) and scalability (peak tenants processed per minute). Resource usage is monitored through monitoring over-provisioning (idle resources more than 20% of capacity) and under-utilization (too-low resources at less than 15% utilization).

Table 2: Model Training Efficiency

Model	**Training Time (hours)**	**GPU Memory (GB)**
LSTM	4.2	8
Transformer	8.9	16
LSTM-Transformer	6.5	12

4. Experimental Setup and Implementation

4.1 Simulation Environment

The test environment is constructed on top of a Kubernetes cluster that runs 50 AWS EC2 c5.4xlarge instances, 16 vCPUs, 32 GB RAM, and NVIDIA T4 GPUs each. The cluster has a simulated multi-tenant workload generator running that simulates heterogeneous tenant behaviors such as bursty traffic, periodic cycles, and stochastic demand patterns. Synthetic traces are constructed with a combination of sinusoidal base patterns (to mimic daily/weekly cycles) and superimposed Gaussian noise (σ=15%) to model real-world volatility. To evaluate in the real world, anonymized workload traces from the 2018 dataset are used, comprising 4,000 tenants with metrics of CPU, memory, and network I/O sampled at every five minutes. Tenants are isolated through Kubernetes namespaces and quota on resources, and network policies limit inter-tenant communication to simulate production environments.

4.2 Tools and Frameworks

Forecasting pipeline AI uses TensorFlow 2.8 for LSTM implementation and PyTorch 1.12 for transformer components, optimized by mixed-precision training, which saves GPU memory by 30%. Apache Airflow automates workflows end-to-end, including data ingestion (from Kafka topics), preprocessing (Pandas and NumPy), and daily retraining of a model that's called daily at UTC midnight. Kubernetes custom resource definitions (CRDs) are used to run automation scaling policies, which communicate through the forecasting API over RESTful endpoints. Monitoring is performed by Prometheus and Grafana, that watch metrics like pod latency, node usage, and API error rates in real-time.

4.3 Baseline Comparisons

Benchmark models are ARIMA (p=2, d=1, q=1), Holt-Winters exponential smoothing (triple seasonality), and XGBoost (500 trees, max depth=6). All are trained with the same data in a 70-30 train-test split on ARIMA inputs, seasonal decomposition. Legacy ML models utilize the same feature sets as the AI framework for a comparable comparison. Performance metrics are run with the same hardware configuration, with autoscaling of Kubernetes

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

nodes turned off to disentangle forecasting quality from infrastructure behavior.

Table 3: Synthetic vs. Real-World Dataset Characteristics

Metric	Synthetic Data	Real-World Data
Tenants	1,000	4,000
Sampling Interval	5 minutes	5 minutes
Noise Level	15%	N/A (Anonymized)
Peak Traffic	2,000 RPS	3,500 RPS

5. Results and Analysis

5.1 Forecasting Accuracy

The hybrid LSTM-Transformer model performs best among all baseline methods on all forecasting accuracy metrics. On synthetic data, the model has an RMSE of 6.7 and MAPE of 14.2%, a gain of 34% over ARIMA (RMSE=12.4, MAPE=28.6%) and 22% over simple LSTM (RMSE=8.2, MAPE=18.3%). The transformer's self-attention is able to effectively catch cross-tenant correlations, e.g., holiday sale induced spikes in correlated demand between retail tenants, and cuts 19% of prediction errors for high-variability workload scenarios. On actual data, model robustness is preserved with the MAPE only increasing by modest amounts to 15.8% with unmodeled workload anomalies, e.g., infrastructure failures not included in the synthetic data.

Table 4: Forecasting Accuracy Across Workload Types

Workload Type	RMSE	MAPE (%)	Improvement vs. ARIMA
Bursty (Video)	9.1	17.5	29%
Periodic (SaaS)	5.3	12.8	38%
Stochastic (IoT)	7.8	16.2	26%

5.2 Capacity Planning Efficiency

AI-based capacity planning lessens over-provisioning from 35% to 21% and under-utilization from 28% to 12% within the Kubernetes cluster. Policy-based automated scaling out nodes in 800 ms of forecast demand levels, reducing transient SLA violations by 73% compared to reactive practices. For a 10,000 tenant test SaaS application, that is an annual saving of $8.2 million if average cloud instance cost per hour is $0.10. The system scales dynamically to workload changes, such as a 300% surge in traffic for video streaming tenants in prime time, by pre-provisioning the GPU resource 15 minutes in advance.

5.3 Scalability Analysis

It linearly scales to 1,000 tenants with end-to-end latency going up from 220 ms (100 tenants) to 920 ms (1,000 tenants). Throughput drops by 53% at 1,000 tenants due to GPU memory contention by processing 210 requests per second versus 450 requests per second for 100 tenants. Yet, the distributed training component of the hybrid architecture eases this with horizontal pod autoscaling by spawning LSTM-Transformer replicas during high loads. Memory per tenant is locked at 12 MB to enable realistic deployment on resource-limited edge nodes.

5.4 Real-Time Processing Capability

The system records a 95th percentile end-to-end latency of 780 ms and meets sub-second real-time decision-making requirements. Predictions are cached for tenants with stable demand patterns (e.g., <5% hourly utilization change), eliminating redundant computation by 40%, with capacity reserved for GPU cycles for transient workloads. Kubernetes API response time is less than 200 ms during autoscaling as well, enabling smooth integration into production deployments.

Table 5: Cost-Benefit Analysis (10,000 Tenants)

Metric	Traditional System	AI-Driven System
Annual Cost	$20.5M	$12.3M
SLA Violations	12%	3%
Manual Interventions	1,200/year	60/year

6. Discussion

6.1 Interpretability of AI Models

Interpretability of AI-based forecast models is still a problem in multi-tenant contexts, as operational personnel need to see what is happening for confidence in automated decisions. The hybrid LSTM-Transformer model offers some interpretability in terms of attention maps, which specify correlations between tenant workloads and external parts. For

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

instance, attention weights depict retail tenants with evening peak hours (6–10 PM local time), enterprise tenants with 9 AM when business is in session. Transformer layer complexity hides understated feature contribution, and it is difficult to spot outliers, like spikes in latency due to unforeseen network congestion. Methods such as SHAP (SHapley Additive exPlanations) values will enhance explainability but incur computational cost, adding 120 ms of prediction delay per tenant. Accuracy and explainability have to be balanced with care, especially in regulated markets where audit trails are required.

6.2 Practical Implications

The suggested framework has real-world advantages for cloud hosts and SaaS providers by saving annual infrastructure expense by 40% for mid-scale deployments. For example, a 10,000-renter system generates $8.2 million in annual savings through reduced over-provisioning and SLA fines. Kubernetes integration allows effortless adoption to prod environments, where already 92% of organizations employ container orchestration tools. Moreover, the number of manual interventions—12 per day to less than 1—allows DevOps engineers to allocate time to strategic projects. But migration from existing systems is followed by initial investments in GPU hardware and employee training, and estimated ROI time of 14 months for companies with 500+ tenants.

6.3 Limitations and Trade-offs

The only restriction of the framework is the utilization of GPU resources, where the transformer section uses 16 GB of memory per 1,000 tenants. This limits edge deployments, where nodes have 8 GB or less. Reducing model sophistication by pruning or quantization lowers the accuracy of forecasts, raising MAPE by 4–6%. The second trade-off happens in cold-start situations: transfer learning speeds up onboarding for new tenants, yet their first few predictions for tenants with histories of less than seven days have a 22% higher RMSE than settled tenants. In addition, the system also favors scalability over granularity, predicted ahead of every five minutes instead of real-time streams, potentially missing sub-minute demand patterns in high-frequency trading or IoT scenarios.

6.4 Future Research Directions

Future work will need to give highest priority to federated learning platforms to facilitate privacy-friendly model training for tenants, overcoming GDPR and CCPA compliance issues. Integration with Edge-AI would decentralized forecasting activities, lowing latency by local processing and aggregation in regional hubs. Experiments on neuromorphic computing or quantum annealing would further lower energy demands because the existing paradigm takes 1.2 kWh to make 1,000 predictions. Besides, light-weight models such as distilled transformers or temporal convolutional networks (TCNs) would lower GPU memory usage in return for accuracy. Last, integrating causal inference models would bring in resistance to external shocks, i.e., global outages or cyberattacks, which present models come in the form of noise.

7. Conclusion

This study validates the revolutionizing impact of AI-driven demand forecasting for solving complexity of capacity planning in multi-tenant systems. By combining hybrid LSTM-Transformer architectures with transfer learning, the produced framework presents a 34% accuracy improvement in forecasting (RMSE) and a 40% over-provisioning cost savings compared to traditional statistical approaches. The scalability of the model to 1,000+ tenants at sub-second latency makes it significant to dynamic cloud and SaaS domains, where real-time decision-making is essential. Key technical innovations are tenant-specific feature engineering design approach, policy automation using Kubernetes, and a transfer learning pipeline that decreases new tenant onboarding by 65%.

The architecture carries real-world implications that are meaningful, providing a 10,000-tenant SaaS service with annual cost savings of $8.2 million while decreasing SLA breaches by 73%. Strategic investments in GPU capacity and staff training are necessary for successful implementation, however. Rollouts must initially focus on incremental, non-production workload rollouts to validate model performance prior to widespread rollout into production. Further innovation in federated learning and edge-AI deployment can further improve privacy, decrease latency, and avoid GPU resource pains, especially for edge rollouts.

Briefly put, this work bridges the essential divide between predictive analytics and operational effectiveness in multi-tenant environments. Combining scalability, accuracy, and real-time responsiveness, the framework forms a solid foundation for next-generation capacity planning, which enables enterprises to optimize resource utilization while keeping themselves mindful of changing regulatory and operational needs.

References

[1] Alhamazani, M., Chen, L., & Sinnott, R. O. (2011). Business intelligence in the cloud. In 2011 International Conference on Cloud and Service Computing (pp. 1-6). IEEE. https://doi.org/10.1109/CSC.2011.6138510

[2] Alshehri, M. A., Sinnott, R. O., & Chen, L. (2013). Business intelligence as a service for cloud-based applications. In 2013 IEEE International Conference on Big Data (pp. 1-6). IEEE. https://doi.org/10.1109/BigData.2013.6691659

[3] Alhamazani, M., Chen, L., & Sinnott, R. O. (2012). Taking the business intelligence to the clouds. In 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (pp. 645-652). IEEE. https://doi.org/10.1109/TrustCom.2012.285

[4] Brogi, A., Forti, S., & Ibrahim, A. (2010). Cloud broker: Bringing intelligence into the cloud. In 2010 Fourth International Conference on Digital Society (pp. 54-61). IEEE. https://doi.org/10.1109/ICDS.2010.25

[5] Pokorny, J. (2012). How cloud computing is (not) changing the way we do BI. In 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops (pp. 14-19). IEEE. https://doi.org/10.1109/COMPSACW.2012.10

[6] Wang, Y., Wang, Y., & Li, Z. (2011). Cloud computing based business intelligence platform and its application in the field of intelligent power consumption. In 2011 International Conference on Cloud and Service Computing (pp. 7-12). IEEE. https://doi.org/10.1109/CSC.2011.6138511

[7] Tzivara, A. K., Kostopoulos, G. V., & Varvarigou, T. A. (2012). Incorporating business intelligence in cloud marketplaces. In 2012 IEEE 11th International Symposium on Network Computing and Applications (pp. 253-256). IEEE. https://doi.org/10.1109/NCA.2012.20

[8] Alhamazani, M., Chen, L., & Sinnott, R. O. (2020). Serverless architecture for big data analytics. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 1-8). IEEE. https://doi.org/10.1109/BigData50022.2020.9378343

[9] Alhamazani, M., Chen, L., & Sinnott, R. O. (2014). Cloud BI: Future of business intelligence in the cloud. *Journal of Computer and System Sciences, 80*(1), 520-533. https://doi.org/10.1016/j.jcss.2014.06.013

[10] Chang, V. (2014). The business intelligence as a service in the cloud. *Future Generation Computer Systems, 37*, 512-534. https://doi.org/10.1016/j.future.2013.12.028

[11] Sheshasaayee, A. (2015). The challenges of business intelligence in cloud computing. *Indian Journal of Science and Technology, 8*(36), 1-6. [No DOI]

[12] Hellerstein, J. M., Gonzalez, J. E., Shute, J., & Stoica, I. (2019). The rise of serverless computing. *Communications of the ACM, 62*(11), 42-51. https://doi.org/10.1145/3368454

[13] Chang, V., Wills, G., & Walters, R. J. (2012). Cloud computing and business intelligence. *International Journal of Cloud Computing, 1*(1), 17-34. https://doi.org/10.1504/IJCC.2012.046546

[14] Alhamazani, M., Chen, L., & Sinnott, R. O. (2014). Cloud-based business intelligence: A review. *International Journal of Cloud Computing, 3*(2), 113-128. https://doi.org/10.1504/IJCC.2014.063247

[15] Ahmed, A. A. A., Donepudi, P. K., & Asadullah, A. B. M. (2020). Opportunities and challenges of data migration in cloud. *Engineering International, 8*(2), 1-12. https://doi.org/10.18034/ei.v8i2.529