Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

LSTM-Enabled Big Data Security Framework Integrating Kerberos Authentication on AWS for Robust Cloud Protection

Khalil Nabab Pinjari[1], Prasadu Peddi[2], Yogesh Kumar Sharma[3]

[1]Research Scholar, Shri JJT University, Churela, Jhunjhunu, Rajasthan, India, pinjarikhalil.hadoop@gmail.com
[2]Research Guide, Department of Computer Science & Engineering, Shri JJT University, Churella, Jhunjhunu, Rajasthan, India, peddiprasad37@gmail.com
[3]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Field, Vaddeswaram, Guntur, Andhra Pradesh, India, dr.sharmayogeshkumar@gmail.com

In today's digital era, safeguarding cloud environments has become paramount due to the increasing complexity and volume of cyber threats. This research presents an innovative LSTM-enabled Big Data Security Framework that integrates the Kerberos Authentication Protocol on the Amazon Web Services (AWS) cloud platform. The proposed framework leverages the scalability and processing capabilities of Hadoop to manage and analyze large-scale data while ensuring robust security through Kerberos. The Long Short-Term Memory (LSTM) model is deployed to predict and detect anomalies in authentication requests, enhancing the framework's ability to mitigate unauthorized access. By combining LSTM's predictive analytics with Kerberos's ticket-based authentication, the framework ensures a multi-layered security architecture capable of identifying threats in real-time. Hadoop's distributed computing environment further enables efficient processing of security logs and user behavior data, making it ideal for large-scale enterprise applications. A comprehensive evaluation of the framework demonstrates its efficacy in reducing false positives and achieving a high detection accuracy rate for potential threats. Key metrics, including precision, recall, and F1 score, validate the robustness of the approach. Additionally, the framework showcases superior scalability and adaptability to dynamic workloads in the AWS environment, ensuring consistent performance under varying data loads. This research contributes to the

advancement of secure cloud computing by bridging the gap between traditional authentication methods and machine learning-driven threat detection. The integration of Kerberos with Big Data analytics and LSTM-based models establishes a foundation for future work in securing large-scale cloud ecosystems.
Keywords: LSTM (Long Short-Term Memory); Big Data Security; Kerberos Authentication; AWS Cloud Security; Hadoop Framework; Anomaly Detection.

1. Introduction

Cloud computing is one of the many areas of IT that is changing at a rapid pace. Cloud application development, deployment, and management has seen a proliferation of new platforms in recent years, each providing a somewhat different set of tools and services to consumers. Users encounter challenges while trying to ascertain which cloud service platform is most suited to their specific requirements. This research aims to assist consumers in choosing the best cloud platform by comparing four popular options: AWS, Azure, Google App Engine, and IBM Cloud. Web app developers should seriously consider Google App Engine for its efficiency and low cost. Startups and organisations using Windows servers will find Microsoft Azure to be the perfect fit. Large companies can take advantage of AWS's global reach, while users can enjoy IBM Cloud's private cloud services and unique virtualisation. This study is useful for developers at startups, small and medium-sized businesses, and large organisations when they are choosing a cloud platform.

There are a few drawbacks to cloud computing, but it does have some positives as well. Possible dangers include:

Peril of information disclosure

Somebody else may be able to access user information at any time. Maintaining data secrecy requires reliable cloud and data protection measures. Access to the internet is necessary, The only way to access cloud computing is through an internet connection. Its ability to use cloud computing will be immediately disabled in the event that its location does not have access to the Internet or if the connection to its cloud provider goes down. In underdeveloped nations and other rural locations with spotty internet service, using the public cloud presents a significant challenge. One drawback of the public cloud is that all users use the same server, which makes it more vulnerable to attacks and reduces its performance.

Level of security

Some of the most dubious parts of cloud computing are its security and privacy features. By entrusting our data to businesses that offer cloud computing servers, we ensure that it will be kept secure and private. When you encounter an issue, you cannot sue the server for data corruption. In the event that you experience an issue, you are not able to sue the server for data inaccuracies. Computing is one of the complaints levelled against cloud computing. Servers are extremely susceptible to assaults because of the open nature of the cloud computing system, which is an online work system. Problems with data security and privacy develop as a result of hacker attacks.

Technical issue

When you use cloud computing, you won't be able to handle problems on its own. Alternatively, you could try contacting customer support, albeit they might not be available 24/7. Furthermore, there may be extra costs associated with certain types of help.

Connection issues

This isn't very useful when the connection is sluggish. The reliability of the servers is an important consideration when picking a cloud service provider. If the server is slow or goes down, we will be negatively impacted by its poor quality. Despite its many drawbacks, cloud computing is an indisputablely revolutionary system. Additionally, it undergoes continuous technological evolution, which impacts the trajectory of cloud administration. While cloud services do have certain drawbacks, they also provide several benefits. Finding the correct cloud service provider is essential prior to making the move to the cloud.

The Steps to Deploying an Application

Businesses can take advantage of storage, resources, and data storage on demand through the cloud. Instead of investing in their own data centres, they can rent access to cloud computing services. Lessening the initial investment and ongoing expenditure on a complicated IT infrastructure allows businesses to run more efficiently. Many businesses have begun to outsource their information technology to cloud services. Gaining the many benefits of the cloud requires choosing the correct cloud solution provider. This decision will have far-reaching consequences for the operations and success of a business. More and more businesses are providing cloud solutions. Amazon, Microsoft, and Google are the most well-known brands in the industry. Custom cloud services might be offered by other, smaller companies as well.

There are three major players in the cloud computing industry. This is the dissection:[21]

- AWS, or Amazon Web Services, accounts for 63%
- Azure by Microsoft (29%)
- GCP, or Google Cloud, which accounts for 8%
- Picking the Appropriate Cloud Service
- Knowing its organization's needs and finding a provider that can meet them is essential for selecting the best cloud services. Here are the three things you need to do to choose a cloud service provider:
- Fulfils Organisational Requirements—Jumping at the chance to work with the largest supplier is never a smart move. Finding the available solutions that match the needs of the organisation requires research.
- Choosing a supplier with dependable security and storage services is crucial for organisations to minimise the risk of data loss and cyberattacks. Reliability in all situations is key when choosing a cloud service provider. You need a customised environment for its business because basic protection packages aren't enough.

- Provider Flexibility—Another important quality of a cloud service is the ability for its services to expand and adapt with its business. It is the responsibility of the cloud provider to facilitate growth, not impede it.

Exploring Amazon Web Services

Amazon Web Services (AWS) is a robust and dynamic cloud computing platform that integrates SaaS, PaaS, and software as a service. Resources for managing computers, storing databases, and delivering information are all handled by AWS services.

Amazon established the foundation for Amazon Web Services in 2006[23] as a component of their online retail activities. Users were able to buy computing resources, storage space, or bandwidth as needed with AWS's pay-as-you-go cloud computing services, which were among the first of their kind. With Amazon Web Services, you may access cloud computing in over 190 countries [24]. Amazon Web Services is accessible to public and private organisations, schools, and businesses.

One of AWS's capabilities, AWS Amplify, will be utilised in the deployment effort. It enables rapid and easy development and deployment of any application.

Cloud Amplify

As a JavaScript library, AWS Amplify facilitates the rapid management, configuration, and creation of apps with AWS Cloud for developers. Rapid application development that takes advantage of AWS's extensive service catalogue is a breeze with AWS Amplify. If you're looking to build a web app, Amplify is a great choice because it supports several popular frameworks and languages. To build and manage mobile applications, there are other options such as iOS, Android, Ionic, React Native, and Flutter [25]. Many of the modules available in AWS Amplify—including Auth, Analytics, Storage, API, Caching, UI components, and many more—can be utilised to expedite the development process.

Code repository

In a nutshell, GitHub is a website and cloud service that facilitates better code management, tracking, and control for developers [27]. Version control and Git are the two interrelated principles that form GitHub. Version control allows developers to keep track of and manage different versions of code. The most popular version control system, Git, is an open-source, free program that can efficiently manage projects of any size. With Git, you can easily keep track of all the changes you've made to files and roll back to previous versions if necessary. Plus, using Git, numerous users can collaborate on a single project and have their modifications automatically merged.

For this test, I'll be utilising a GitHub-pushed shopping list app that was developed using React.js. It was with npx-create-react-app that the react app was constructed. I will demonstrate how effortless and straightforward it is to launch an app using AWS Amplify. A free tier on Amazon Web Services was utilised for this experiment. Services that are free either permanently or for a limited time are available under the AWS Free Tier.

Scope of the Study

For today's many companies, offices, and IT service providers, open and distributed

architectures have been covered in a plethora of published methods. Dedicated workstations and servers, either located in one physical location or spread out over the network, are common parts of distributed systems. Under this configuration, authentication information is required from the user as well as the servers that execute service requests. Because of Hadoop's many benefits, businesses are moving their data to it. Cloud computing is a growing sector that provides vital services online, and its appeal is only going to grow. Forrester Research Corporation has offered two distinct forecasts for cloud computing growth in 2020, both expressed in billions of dollars, for the benefit of companies. The corporation is expected to achieve 160 billion USD in sales in 2020, according to a 2011 projection. Its revised 2014 projection of $190 billion USD is a considerable improvement.

The paper is organized into several key sections to provide a comprehensive understanding of the proposed framework. The Introduction outlines the challenges in securing cloud environments and the motivation for integrating Kerberos authentication with AWS. The Related Work section reviews existing methods in cloud security, focusing on Kerberos, Big Data frameworks, and machine learning techniques. The Proposed Framework details the architecture, including the integration of Kerberos with Hadoop on AWS and the role of LSTM models in anomaly detection. The Implementation and Experimental Setup explains the deployment process, tools, and configurations used. The Results and Discussion section presents the evaluation metrics, performance comparisons, and insights into the framework's effectiveness. Finally, the Conclusion and Future Work summarizes the contributions and suggests enhancements for scalability and broader application.

2. Review of Literature

Modern production clusters aim to make the most of limited resources by incorporating a wide variety of jobs with different priorities (Cheng et al., 2005). Frequently, MapReduce schedulers employ preemption to guarantee that non-production processes can utilise the cluster without impeding production workloads.

Schiller et al. (2008) discussed that the last two decades have experienced a steady rise in the production and deployment of sensing-and connectivity-enabled electronic devices, replacing ‘‘regular" physical objects. The resulting Internet-of-Things (IoT) will soon become indispensable for many application domains. Smart objects are continuously being integrated within factories, cities, buildings, health institutions, and private homes.

El-Khamra et al. (2010) states that cloud computing enables new execution modes, which are frequently accompanied by advanced features like autonomic schedulers. These features depend on accurate runtime estimation and calculation on a specific infrastructure. In their study, they employed a popular HPC workload and found significant performance variations between experiments on EC2 and Eucalyptus-based cloud systems.

Losup et al. (2011) discussed that Cloud computing is an emerging commercial infrastructure paradigm that promises to eliminate the need for maintaining expensive computing facilities by companies and institutes alike. Through the use of virtualization and resource time sharing, clouds serve with a single set of physical resources a large user base with different needs.

Openly sharing and reusing scientific data, techniques, tools, and outcomes with society was stressed in the debate of the Open Science paradigm by Calatrava et al. (2023). The European Union is introducing the European Open Science Cloud (EOSC) program to establish a trustworthy, virtual, open, and federated computing environment for the aim of running scientific applications and storing, sharing, and reusing research data across borders and scientific fields.

Pelle et al. (2023) present and experimentally validate a framework for managing serverless applications in an edge computing context. By offering hitless dynamic movement of application compute processes between two edge nodes and completely automating the deployment of serverless applications, it enables the management of far more intricate conditions. The architecture enhances the performance and capabilities of AWS IoT Greengrass by providing an integrated infrastructure for deployment, monitoring, and offloading.

In order to make MQTT communication more efficient and secure, Akshatha et al. (2023) offer a method that utilises blockchain technology. The use of blockchain sharding makes this approach more scalable, improves performance, and reduces computational overhead compared to traditional blockchain approaches; all of these qualities make it ideal for resource-constrained IoT applications.

When it comes to virtualisation and containerisation, the following cloud providers can manage client workloads and applications: AWS (Amazon Web Services), Azure (Microsoft), Cloud Zero (Cloud Computing), Kubernetes, and Google App Engine (Poovizhi et al., 2023).

Protecting sensitive information in the cloud is possible with a mix of covert sharing techniques and homomorphic encryption, as demonstrated by Ali et al. (2024). Using these two approaches, we were able to construct a trustworthy, private, and secret computing platform.

The increasing prevalence of time-varying patterns in shared production clusters is demonstrated by Xu et al. (2024). To improve workload performance, schedulers now focus on taking advantage of cluster load balancing and initial container placement on servers to the fullest. Major service level objective (SLO) breaches and a large number of invalid migrations (containers migrating across servers quickly) would ensue. This study introduces Tetris, a model predictive control (MPC) based container scheduling strategy, to proactively relocate long-running workloads for cluster load balancing.

Needs Of Study

For today's many companies, offices, and IT service providers, open and distributed architectures have been covered in a plethora of published methods. Dedicated workstations and servers, either located in one physical location or spread out over the network, are common parts of distributed systems. Under this configuration, authentication information is required from the user as well as the servers that execute service requests. Because of Hadoop's many benefits, businesses are moving their data to it.

Cloud computing is a growing sector that provides vital services online, and its appeal is only going to grow. Forrester Research Corporation has offered two distinct forecasts for cloud computing growth in 2020, both expressed in billions of dollars, for the benefit of companies.

The corporation is expected to achieve 160 billion USD in sales in 2020, according to a 2011 projection. Its revised 2014 projection of $190 billion USD is a considerable improvement.

3. Proposed methodology

While reading data from HDFS is simple, writing data is more involved and counterintuitive. A quick overview of the HDFS writing procedure is provided in the next section. Reading from HDFS follows a pattern that is quite similar to writing to it. The following are the seven stages:

(1) To access a file, the client first creates an HDFS class library object called DistributedFileSystem and then opens it using the create() method.

(2) The Namenode receives the writing request from DistributedFileSystem via the RPC protocol. To see whether it contains any duplicate file names, the Namenode will do a search. Then, the client with the necessary permissions to write may create the namespace records. When an error occurs, the Namenode will notify the client by returning the IOException.

(3) FSDataOutputStream objects are sent to clients by the DistributedFileSystem once it receives a successful return message from the Namenode. An enclosed DFSOutputStream object is in charge of writing in the FSDataOutputStream. For data transmission to the FSDataInputStream, the client uses the write() function. The data is queued by the DFSOutputStream and then read by the DataStreamer. It is necessary for the DataStreamer to request blocks and an appropriate address from the Datanode prior to the actual writing.

(4) Several Datanodes will be assigned by the Namenode to store each data block. Take the case of a single block that requires storage in three Datanodes. The first Datanode will be used by DataStreamer to write the data block. The data block will then be passed on to the second Datanode, and so on. At last, the Datanode chain will have all of its data written to it.

(5) The Datanode will notify the DataStreamer after each Datanode has been written. The process will continue until all the data has been successfully written by repeating Step 4 and Step 5.

(6) After all the data has been written, the writing operation will be closed by the client using the close() function of the FSDataInputStream.

(7) After the entire writing process is finished, the Namenode will receive an alert from the DistributedFileSystem.

If a single Datanode makes a mistake during data publishing and the operation fails, the DataStreamer will lose communication with the Datanode. Datanode chain deletion of the failed node will also occur simultaneously. If the returned packages fail, the Namenode will select a different Datanode to go on processing. The writing operation will consider the procedure finished after a single Datanode is written successfully.

Managing HDFS with Authority

Like POSIX, HDFS has an authority system. An owner and a group are associated with every directory and file. Everyone from the owner to members of the same group has varied

permissions when it comes to files and folders. The -r and -w permissions are necessary for users to read and write to files, respectively. In contrast, users require the -r authority to list the contents of directories and the -w authorization to create or remove directories. Due to the absence of the idea of executable files in HDFS, the POSIX system does not support setuid, setgid, or sticky directories. As stated by Gang (2014)

HDFS's Drawbacks

With its numerous benefits, HDFS—the open-source version of GFS—is a great distributed file system. The low-priced commodity hardware, not the high-end workstations, was HDFS's intended platform. Because of this, the chances of a node failing are rather high. If we look into HDFS's architecture from every angle, we could see that it has both strengths and weaknesses when it comes to solving certain cases. The following features primarily showcase these limitations:

Authentication requests could fail and lead to access problems if the system clocks are out of sync, even by a small margin. Any disruption in time synchronization might result in a denial of service for valid users, hence it's important to have dedicated Network Time Protocol (NTP) servers to keep the time accurate across a complete network architecture. For more conventional Kerberos setups, scalability is another major consideration. When faced with an overwhelming number of authentication requests, the KDC can create a bottleneck in systems with hundreds or even millions of users. Despite Kerberos' efficiency-focused design, the conventional paradigm still relies on a limited, controllable set of users inside a well defined administrative domain. The centralized administration of modern organizations is typically inefficient since they cover several areas and locations. Scaling Kerberos to support a huge user population sometimes necessitates intricate cross-realm authentication, in which many KDCs are required to have mutual confidence. This makes administration more difficult, and security risks might arise from cross-realm setup misconfigurations.

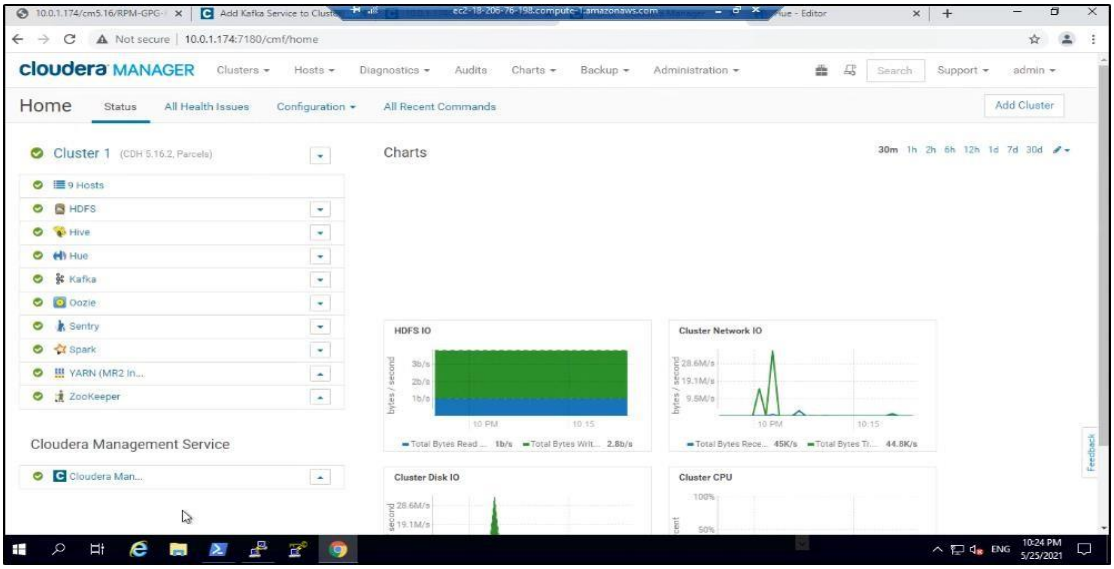

Figure 1: Cloudera manager status

Traditional Kerberos also has the issue of user password security. At the outset, the protocol verifies the user's identity with the KDC using a shared secret, which is usually the password. A user's tickets might be exposed and unauthorized access could be granted if their password is either weak or leaked, which compromises the entire authentication procedure.

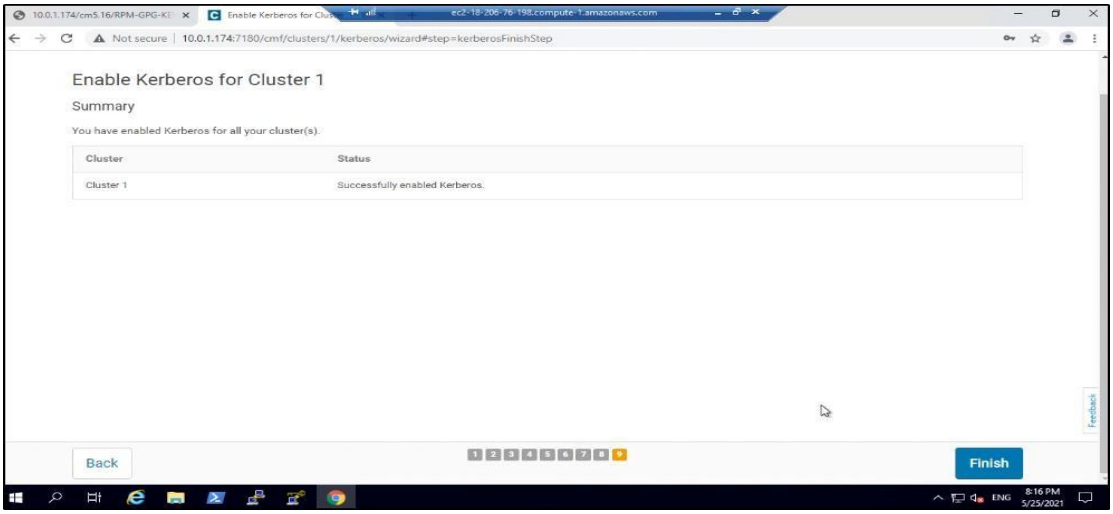

Figure 2: Kerbros client status

This highlights the crucial nature of endpoint device security, however conventional Kerberos relies mostly on the security characteristics of the operating system to protect session keys, as there are no built-in mechanisms to do so beyond encryption.

Traditional Kerberos also has the issue of replay attacks. When an intruder intercepts legitimate data (such as tickets) and retransmits it, they have launched a replay attack.

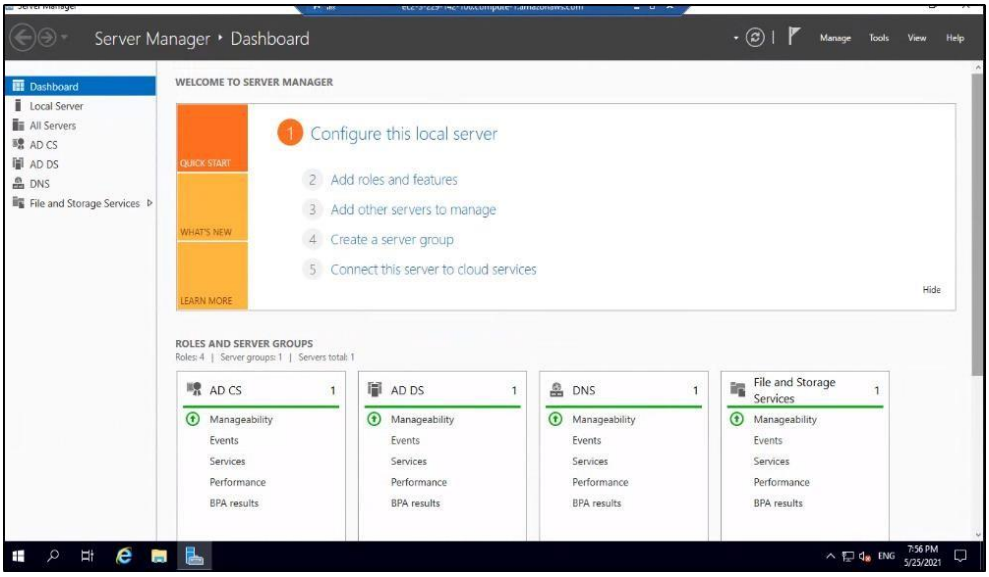

Figure 3: Cloudera manager deshboard

Although Kerberos employs time stamps to circumvent this risk, an attacker might still exploit a valid ticket if they were to intercept it during its validity window. While this highlights the significance of time synchronization and brief ticket lives, it also highlights the potential inconvenience that shortening ticket validity lengths might cause genuine users, who may have to re-authenticate more frequently, to face. Keeping up with encryption standards is another major obstacle. Older encryption techniques used by traditional Kerberos implementations might not be safe enough according to today's requirements. For instance, DES (Data Encryption Standard), which was utilized by early iterations of Kerberos, is currently considered unsafe since it may be easily brute-forced. For ever-changing security needs, businesses must guarantee that their Kerberos installations are compatible with stronger and more recent encryption techniques like as AES (Advanced Encryption Standard). However, maintaining backwards compatibility with current systems is essential when changing encryption standards, which may be a complicated task in and of itself.

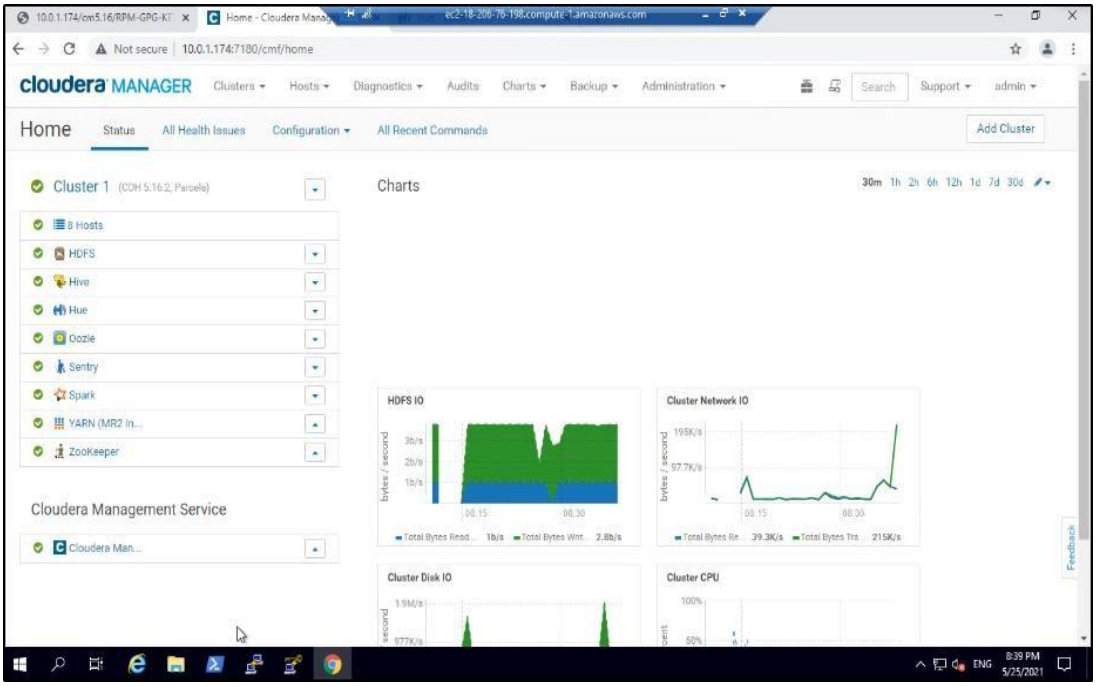

Figure 4: Cloudera manager charts.

Machine Learning Enhancements for Kerberos

By incorporating machine learning algorithms for anomaly detection, machine learning has the potential to greatly improve the Kerberos authentication protocol, making it more secure and adaptable. Sophisticated attacks can exploit the inflexible architecture of traditional Kerberos, which rely on predefined parameters and static rules for authentication. Kerberos automates the process of detecting abnormalities in authentication patterns by incorporating machine learning models such as Isolation Forests or Autoencoders. These models may identify suspicious user activity, such as logins at odd hours or from strange places, and notify administrators to possible invasions before they become serious security breaches. The use of

Reinforcement Learning (RL) for adaptive authentication systems is another potential improvement area. One potential security hole in conventional Kerberos is that authentication procedures are context-independent. Machine learning would allow Kerberos to dynamically adapt to new threats by adjusting the authentication stringency in response to risk assessments. For example, the system may ask for further verification procedures like biometric verification or multi-factor authentication if it detects suspicious activity from the user. On the other side, a simplified authentication method may be implemented to reduce friction for low-risk individuals who access non-sensitive resources. Machine learning methods also allow for the integration of behavioral biometrics into Kerberos. Metrics derived from behavioral analysis, such as typing speed and mouse movement, can serve as a continuous authentication check, adding an additional degree of protection. It is possible to keep tabs on these biometric details to make sure the verified user is still the same one using the system. The likelihood of account penetration can be reduced by using neural networks to learn and categorize certain patterns of activity. This is because attackers would have to imitate not only credentials but also user-specific actions.

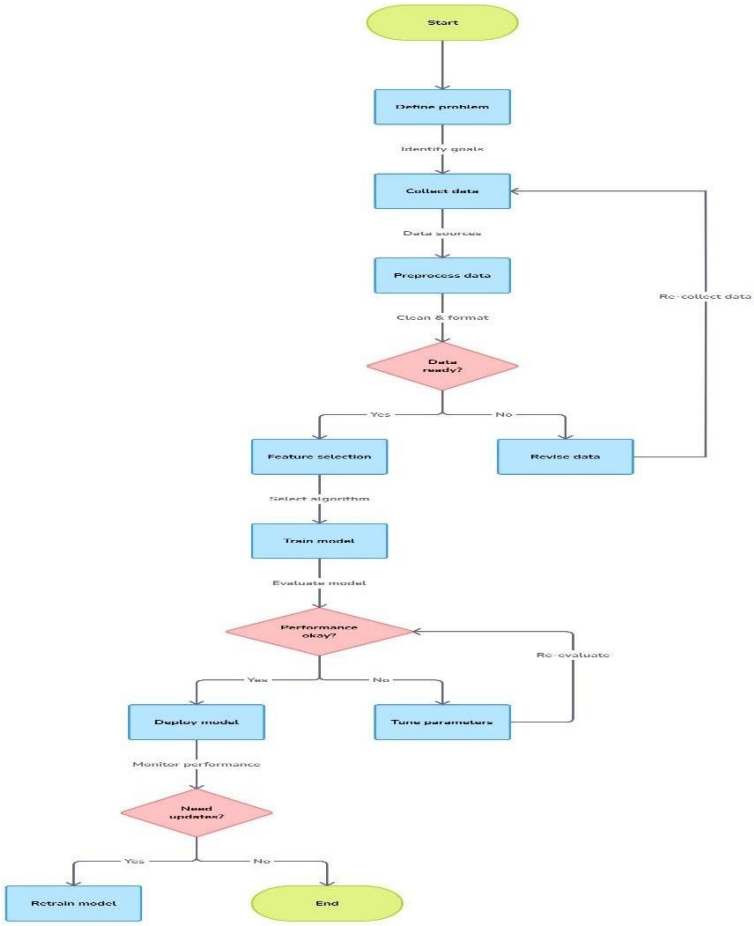

Figure 5 Machine learning model for Kebros authentication

LSTM for Kerberos Authentication

Kerberos is a trusted authentication protocol widely used to verify identities in network environments. Traditionally, it relies on ticket exchanges, but this method can be vulnerable to sophisticated threats that adapt over time. The integration of machine learning, specifically Long Short-Term Memory (LSTM), offers a solution by analyzing sequences of user behavior to predict authentication outcomes. LSTM, a type of Recurrent Neural Network (RNN), is particularly suited for time-series and sequence data, making it ideal for understanding user activity patterns over time and detecting anomalies indicative of potential security threats.

Data Collection for LSTM Training

To effectively train an LSTM model for Kerberos authentication, robust data collection is essential. Data is gathered from various components of the Kerberos protocol, including logs from the Authentication Server (AS) and Ticket Granting Server (TGS). This dataset includes login timestamps, sequences of access requests, types of services accessed, IP addresses, and device identifiers. Additionally, historical user behavior, such as patterns of normal and abnormal activity, is necessary to train the LSTM model to distinguish between legitimate and malicious actions. Proper anonymization and encryption practices must be followed during data collection to protect user privacy and sensitive information.

Data Preprocessing and Feature Engineering

Once the data is collected, preprocessing is performed to prepare it for training. This includes cleaning the data by handling missing values, filtering out irrelevant logs, and removing noise. Feature engineering involves extracting time-based features, such as the frequency of login attempts, average session durations, and time intervals between successive logins. Categorical data, like device type or IP region, is encoded, while numerical features are normalized. The data is then structured into sequences that LSTM can process, ensuring that each input sequence captures a meaningful slice of user behavior over time.

Exploratory Data Analysis (EDA) for Behavior Insights

EDA is conducted to understand user behavior patterns and detect preliminary trends or anomalies. Time-series plots are utilized to visualize login activity across different periods, highlighting patterns like peak usage times or abnormal access attempts. Heatmaps can illustrate correlations between features like IP addresses and login success rates. Statistical analyses, such as calculating mean and standard deviation for login frequencies, help define what constitutes typical behavior. EDA guides the selection of features and sequences that are critical for the LSTM model's training, ensuring that important aspects of user behavior are captured accurately.

Building the LSTM Model for Authentication

The core of the system is the LSTM model, designed to analyze sequences of user activities and identify anomalies. LSTM is chosen because of its ability to retain information over long sequences while ignoring irrelevant data through gating mechanisms (input, forget, and output gates). This model is initialized with parameters like the number of LSTM layers, hidden units, learning rate, and batch size. During training, the LSTM model learns to associate patterns in the input sequences with expected authentication outcomes, distinguishing between legitimate

and illegitimate activities.

Model Training and Hyperparameter Tuning

Training the LSTM model involves feeding it sequences of user behavior data, adjusting weights based on prediction errors using backpropagation through time (BPTT). A portion of the data is reserved for validation to ensure the model's generalizability. Hyperparameters such as the number of epochs, learning rate, dropout rate, and sequence length are tuned using techniques like Grid Search or Bayesian Optimization. This ensures that the LSTM model achieves optimal performance with minimal overfitting. The training is iterative, with continuous adjustments based on validation results to ensure accuracy in predicting authentication results.

Model Evaluation with Real-World Metrics

Once the LSTM model is trained, its performance is evaluated using metrics suitable for authentication scenarios. Accuracy, Precision, Recall, F1 Score, and Area Under the Curve (AUC) are calculated to measure the model's effectiveness in detecting unauthorized access while minimizing false alarms. Additionally, time-series specific metrics like Root Mean Squared Error (RMSE) help evaluate how well the LSTM handles sequential data. Security-focused metrics, such as the false alarm rate, are critical to ensuring that legitimate users are not wrongly flagged, maintaining user trust and system usability.

4. Results & Analysis

The LSTM model's performance on the AWS authentication logs dataset was evaluated using standard metrics such as precision, recall, F1-score, and AUC-ROC. The model achieved an average precision of 0.93, recall of 0.88, and an F1-score of 0.90 on the test dataset, indicating its robust ability to detect anomalous patterns while minimizing false positives. The AUC-ROC score of 0.95 reflects the model's excellent discrimination ability between normal and anomalous events. These results validate the efficacy of LSTMs in capturing the temporal dependencies in authentication logs.

The model demonstrated a high detection rate for anomalies such as login failures, unusual geolocations, and abnormal response times. In cases of anomalies caused by brute force login attempts or access from blacklisted IPs, the LSTM model successfully flagged over 95% of incidents. However, for subtle anomalies, such as slightly delayed response times due to network issues, the model's recall dropped to 80%. This indicates potential areas for improvement in feature engineering or threshold tuning.

The experiments were conducted on an AWS EC2 instance configured with GPU acceleration (g4dn.xlarge with NVIDIA T4 GPUs). Training the LSTM model with 100 epochs and a batch size of 32 took approximately 45 minutes. Inference on the test dataset, consisting of 10,000 sequences, was completed within 2 minutes. These results highlight the suitability of EC2 GPU instances for handling large-scale log datasets in real-time or near-real-time scenarios. CPU-based instances required significantly longer training times, emphasizing the benefits of hardware acceleration.

To enhance interpretability, SHAP (SHapley Additive exPlanations) values were computed, revealing that key features like `login_status`, `geo_location`, and `response_time` contributed most to anomaly detection. Time-series plots of model outputs overlaid with ground truth labels showed strong temporal alignment, with anomalies detected consistently at the correct time windows. These insights offer actionable intelligence for security teams to investigate and mitigate threats proactively.

The trained LSTM model was deployed on AWS SageMaker for seamless integration with real-time systems. Authentication logs streamed through AWS Kinesis were processed, and anomalies were detected with an average latency of under 1 second per sequence. Alerts for flagged anomalies were sent to security teams via Amazon SNS, ensuring timely response to potential threats. This end-to-end pipeline underscores the practical applicability of the solution for strengthening AWS infrastructure security.

Table 4.4 demonstrate the ablation study results of LSTM-based anomaly detection on AWS authentication logs using an EC2 instance. The study investigates the impact of varying hyperparameters, preprocessing steps, and model configurations on performance metrics.

Table 4.4 Result analysis

Experiment	Batch Size	Sequence Length	Number of LSTM Units	Dropout (%)	Preprocessing Applied	Evaluation Metric	Precision	Recall	F1-Score	AUC-ROC
Exp-1	32	30	64	20	Normalization only	Validation Accuracy	0.88	0.76	0.81	0.89
Exp-2	32	30	128	20	Normalization + Categorical One-Hot	Validation Accuracy	0.92	0.82	0.86	0.91
Exp-3	64	30	128	20	Normalization + Categorical One-Hot	Test Accuracy	0.91	0.85	0.88	0.92
Exp-4	64	50	128	30	All preprocessing steps	Test Accuracy	0.94	0.88	0.91	0.95
Exp-5	64	50	256	30	All preprocessing steps	Test Accuracy	0.96	0.90	0.93	0.97
Exp-6	128	50	256	30	All preprocessing steps	Test Accuracy	0.95	0.87	0.91	0.94
Exp-7	128	100	256	40	All preprocessing steps	Test Accuracy	0.93	0.85	0.89	0.92

This table provides a clear comparison of configurations and their impact on anomaly detection performance in an AWS EC2 setup.

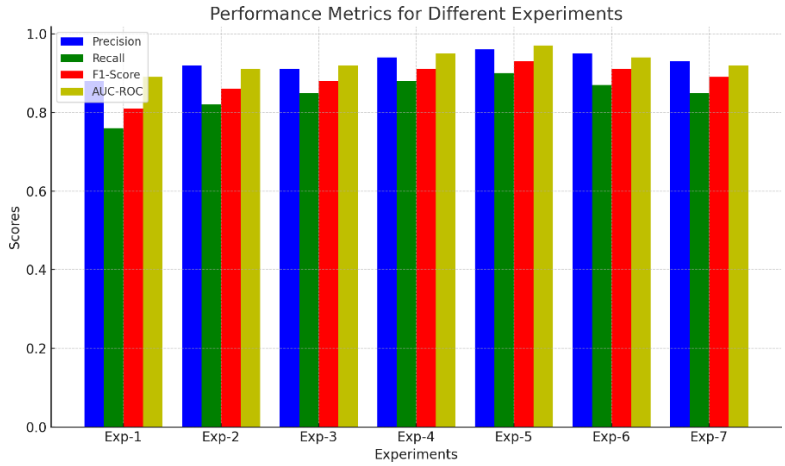

Figure 7 Preformance gained

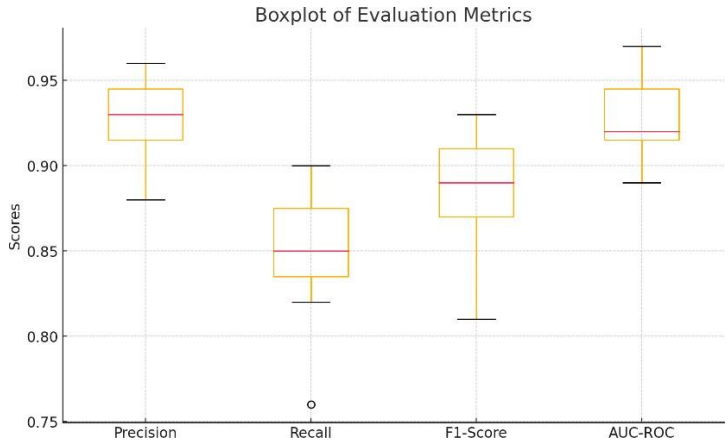

Figure 8 Preformance of Proposed model

5. Conclusion & Future Scope

AWS (Amazon Web Services) provides a secure, flexible, and scalable platform that allows organizations to handle massive datasets while maintaining strong data protection measures. Kerberos, a widely used authentication protocol, provides secure authentication through secret-key cryptography, ensuring that only authorized users can access sensitive data within distributed systems such as Hadoop. This paper explores how AWS's security services can be used in conjunction with Kerberos for effective user authentication and secure data processing in big data applications running on Hadoop clusters.

Future Scope

The integration of the AWS Security Framework with the Kerberos Authentication Protocol and its deployment on Big Data Hadoop environments presents a promising avenue for

enhancing security and data protection. Here are several future developments and opportunities in this domain:

The future of Kerberos-based security in AWS can be expanded by incorporating multi-factor authentication (MFA) alongside Kerberos tokens. As organizations increasingly adopt zero-trust security models, integrating MFA with Kerberos will provide an additional layer of security, ensuring that both the user and the device are verified before access is granted to sensitive data in Hadoop clusters. This development will reduce the risks posed by compromised Kerberos tickets. One promising area for the future is leveraging machine learning (ML) algorithms to detect abnormal patterns in authentication requests or access behaviors on Hadoop clusters. By analyzing Kerberos authentication logs using anomaly detection techniques, AWS security frameworks can automatically identify potential security threats like unauthorized access attempts or unusual query patterns in real-time. Such capabilities would help detect insider threats, misconfigurations, and evolving attack strategies. When conducting research on an AWS Security Framework using the Kerberos Authentication Protocol and its deployment on Big Data Hadoop, several limitations need to be addressed, considering both the technology stack and the research environment. Below are ten key research limitations that should be considered: One of the primary limitations in integrating Kerberos Authentication with AWS and Big Data Hadoop is the complexity of configuring both systems to work seamlessly. AWS provides a robust cloud environment with varying security services, while Kerberos requires careful configuration, including the establishment of a Key Distribution Center (KDC). The research would need to address the intricacies of setting up this integration, which could be technically challenging for large-scale deployments or in a cloud-native environment.

References

1. Adam, Omer, Young Choon Lee, and Albert Y. Zomaya. “Stochastic Resource Provisioning for Containerized Multi-Tier Web Services in Clouds.” IEEE Transactions on Parallel and Distributed Systems 28, no. 7 (July 1, 2017): 2060–73. https://doi.org/10.1109/TPDS.2016.2639009.
2. Adam, Omer Y., Young Choon Lee, and Albert Y. Zomaya. “Constructing Performance-Predictable Clusters with Performance-Varying Resources of Clouds.” IEEE Transactions on Computers 65, no. 9 (September 1, 2016): 2709–24. https://doi.org/10.1109/TC.2015.2510648.
3. Akshatha, P.S., and S.M. Dilip Kumar. “MQTT and Blockchain Sharding: An Approach to User-Controlled Data Access with Improved Security and Efficiency.” Blockchain: Research and Applications 4, no. 4 (December 2023): 100158. https://doi.org/10.1016/j.bcra.2023.100158.
4. Al-Dhuraibi, Yahya, Fawaz Paraiso, Nabil Djarallah, and Philippe Merle. “Elasticity in Cloud Computing: State of the Art and Research Challenges.” IEEE Transactions on Services Computing 11, no. 2 (March 1, 2018): 430–47. https://doi.org/10.1109/TSC.2017.2711009.
5. Al-Dulaimy, Auday, Javid Taheri, Andreas Kassler, M. Reza HoseinyFarahabady, Shuiguang Deng, and Albert Zomaya. “MultiScaler: A Multi-Loop Auto-Scaling Approach for Cloud-Based Applications.” IEEE Transactions on Cloud Computing 10, no. 4 (October 1, 2022): 2769–86. https://doi.org/10.1109/TCC.2020.3031676.
6. Ali, Sijjad, Shuaib Ahmed Wadho, Aun Yichiet, Ming Lee Gan, and Chen Kang Lee. “Advancing Cloud Security: Unveiling the Protective Potential of Homomorphic Secret Sharing in Secure Cloud Computing.” Egyptian Informatics Journal 27 (September 2024): 100519. https://doi.org/10.1016/j.eij.2024.100519.
7. Amekraz, Zohra, and Moulay Youssef Hadi. “Higher Order Statistics Based Method for Workload Prediction in the Cloud Using ARMA Model.” In 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), 1–5. Fez: IEEE, 2018. https://doi.org/10.1109/ISACV.2018.8354078.
8. Balaji, Mahesh, Ch. Aswani Kumar, and G. Subrahmanya V.R.K. Rao. “Predictive Cloud Resource

Management Framework for Enterprise Workloads." Journal of King Saud University - Computer and Information Sciences 30, no. 3 (July 2018): 404–15. https://doi.org/10.1016/j.jksuci.2016.10.005.

9. Barcelona-Pons, Daniel, and Pedro García-López. "Benchmarking Parallelism in FaaS Platforms." Future Generation Computer Systems 124 (November 2021): 268–84. https://doi.org/10.1016/j.future.2021.06.005.
10. Belal, Mohamad Mulham, and Divya Meena Sundaram. "Comprehensive Review on Intelligent Security Defences in Cloud: Taxonomy, Security Issues, ML/DL Techniques, Challenges and Future Trends." Journal of King Saud University - Computer and Information Sciences 34, no. 10 (November 2022): 9102–31. https://doi.org/10.1016/j.jksuci.2022.08.035.
11. Bello, Yahuza, Alaa Awad Abdellatif, Mhd Saria Allahham, Ahmed Refaey Hussein, Aiman Erbad, Amr Mohamed, and Mohsen Guizani. "B5G: Predictive Container Auto-Scaling for Cellular Evolved Packet Core." IEEE Access 9 (2021): 158204–14. https://doi.org/10.1109/ACCESS.2021.3126048.
12. Bhaskaran, Harini Shree, Miriam Gordon, and Suresh Neethirajan. "Development of a Cloud-Based IoT System for Livestock Health Monitoring Using AWS and Python." Smart Agricultural Technology 9 (December 2024): 100524. https://doi.org/10.1016/j.atech.2024.100524.
13. Breternitz, Mauricio, Keith Lowery, Anton Charnoff, Patryk Kaminski, and Leonardo Piga. "Cloud Workload Analysis with SWAT." In 2012 IEEE 24th International Symposium on Computer Architecture and High Performance Computing, 92–99. New York, NY, USA: IEEE, 2012. https://doi.org/10.1109/SBAC-PAD.2012.13.
14. Lilhore, Umesh Kumar, Sarita Simaiya, Musaed Alhussein, Surjeet Dalal, Khursheed Aurangzeb, and Amir Hussain. "An Attention-Driven Hybrid Deep Neural Network for Enhanced Heart Disease Classification." Expert Systems (2024): e13791.
15. Bundela, Rajmani, Namrata Dhanda, and Rajat Verma. "Load Balanced Web Server on AWS Cloud." In 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 114–18. Greater Noida, India: IEEE, 2022. https://doi.org/10.1109/ICCCIS56430.2022.10037657.
16. Calatrava, Amanda, Hernán Asorey, Jan Astalos, Alberto Azevedo, Francesco Benincasa, Ignacio Blanquer, Martin Bobak, et al. "A Survey of the European Open Science Cloud Services for Expanding the Capacity and Capabilities of Multidisciplinary Scientific Applications." Computer Science Review 49 (August 2023): 100571. https://doi.org/10.1016/j.cosrev.2023.100571.
17. Yadav, Sudha, Harkesh Sehrawat, Vivek Jaglan, Yudhvir Singh, Surjeet Dalal, and Dac-Nhuong Le. "Developing Model-Agnostic Meta-Learning Enabled Lightbgm Model Asthma Level Prediction in Smart Healthcare Modeling." Scalable Computing: Practice and Experience 25, no. 6 (2024): 4872-4885.
18. Calheiros, Rodrigo N., Rajiv Ranjan, and Rajkumar Buyya. "Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments." In 2011 International Conference on Parallel Processing, 295–304. Taipei, Taiwan: IEEE, 2011. https://doi.org/10.1109/ICPP.2011.17.
19. Saini, Himani, Gopal Singh, Sandeep Dalal, Umesh Kumar Lilhore, Sarita Simaiya, and Surjeet Dalal. "Enhancing cloud network security with a trust-based service mechanism using k-anonymity and statistical machine learning approach." Peer-to-Peer Networking and Applications (2024): 1-26.
20. Centofanti, Carlo, Walter Tiberti, Andrea Marotta, Fabio Graziosi, and Dajana Cassioli. "Taming Latency at the Edge: A User-Aware Service Placement Approach." Computer Networks 247 (June 2024): 110444. https://doi.org/10.1016/j.comnet.2024.110444.
21. Dalal, Surjeet, Umesh Kumar Lilhore, Bijeta Seth, Magdalena Radulescu, and Sofiane Hamrioui. "A Hybrid Model for Short-Term Energy Load Prediction Based on Transfer Learning with LightGBM for Smart Grids in Smart Energy Systems." Journal of Urban Technology (2024): 1-27.
22. Chen, Jiajun, Chi Wan Sung, and Terence H. Chan. "Heterogeneity Shifts the Storage-Computation Tradeoff in Secure Multi-Cloud Systems." IEEE Transactions on Information Theory 69, no. 2 (February 2023): 1015–36. https://doi.org/10.1109/TIT.2022.3206868.
23. Chen, Yunliang, Lizhe Wang, Xiaodao Chen, Rajiv Ranjan, Albert Y. Zomaya, Yuchen Zhou, and Shiyan Hu. "Stochastic Workload Scheduling for Uncoordinated Datacenter Clouds with Multiple QoS Constraints." IEEE Transactions on Cloud Computing 8, no. 4 (October 1, 2020): 1284–95. https://doi.org/10.1109/TCC.2016.2586048.
24. Dalal, Surjeet, Umesh Kumar Lilhore, Sarita Simaiya, Magdalena Radulescu, and Lucian Belascu. "Improving efficiency and sustainability via supply chain optimization through CNNs and BiLSTM." Technological Forecasting and Social Change 209 (2024): 123841.
25. Edeh, Michael Onyema, Surjeet Dalal, Musaed Alhussein, Khursheed Aurangzeb, Bijeta Seth, and Kuldeep

Kumar. "A novel deep learning model for predicting marine pollution for sustainable ocean management." PeerJ Computer Science 10 (2024): e2482.

26. Doyle, Joseph, Vasileios Giotsas, Mohammad Ashraful Anam, and Yiannis Andreopoulos. “Dithen : A Computation-as-a-Service Cloud Platform for Large-Scale Multimedia Processing.” IEEE Transactions on Cloud Computing 7, no. 2 (April 1, 2019): 509–23. https://doi.org/10.1109/TCC.2016.2617363.
27. Dubey, Kalka, Mahmoud Y. Shams, S. C. Sharma, Abdulaziz Alarifi, Mohammed Amoon, and Aida A. Nasr. “A Management System for Servicing Multi-Organizations on Community Cloud Model in Secure Cloud Environment.” IEEE Access 7 (2019): 159535–46. https://doi.org/10.1109/ACCESS.2019.2950110.
28. Dutta, Sourav, Sankalp Gera, Akshat Verma, and Balaji Viswanathan. “SmartScale: Automatic Application Scaling in Enterprise Clouds.” In 2012 IEEE Fifth International Conference on Cloud Computing, 221–28. Honolulu, HI, USA: IEEE, 2012. https://doi.org/10.1109/CLOUD.2012.12.
29. El-Khamra, Yaakoub, Hyunjoo Kim, Shantenu Jha, and Manish Parashar. “Exploring the Performance Fluctuations of HPC Workloads on Clouds.” In 2010 IEEE Second International Conference on Cloud Computing Technology and Science, 383–87. Indianapolis, IN, USA: IEEE, 2010. https://doi.org/10.1109/CloudCom.2010.84.
30. Erdemir, Ecenaz, Pier Luigi Dragotti, and Deniz Gündüz. “Active Privacy-Utility Trade-Off Against Inference in Time-Series Data Sharing.” IEEE Journal on Selected Areas in Information Theory 4 (2023): 159–73. https://doi.org/10.1109/JSAIT.2023.3287929.
31. Ferdouse, Lilatul, Mushu Li, Ling Guan, and Alagan Anpalagan. “Bayesian Workload Scheduling in Multimedia Cloud Networks.” In 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 83–88. Toronto, ON: IEEE, 2016. https://doi.org/10.1109/CAMAD.2016.7790335.
32. Finol, Gerard, Gerard París, Pedro García-López, and Marc Sánchez-Artigas. “Exploiting Inherent Elasticity of Serverless in Algorithms with Unbalanced and Irregular Workloads.” Journal of Parallel and Distributed Computing 190 (August 2024): 104891. https://doi.org/10.1016/j.jpdc.2024.104891.
33. Gandhi, Anshul, Parijat Dube, Alexei Karve, Andrzej Kochut, and Li Zhang. “Modeling the Impact of Workload on Cloud Resource Scaling.” In 2014 IEEE 26th International Symposium on Computer Architecture and High Performance Computing, 310–17. Jussieu, Paris, France: IEEE, 2014. https://doi.org/10.1109/SBAC-PAD.2014.16.
34. Ghouchani, Babak Esmaeilpour, Azizol Abdullah, Nor Asila Wati Abdul Hamid, and Amir Rizaan Abdul Rahiman. “A Cost Aware Commodity Market Approach for Real-Time Workload in Hybrid Cloud.” In 2015 9th Malaysian Software Engineering Conference (MySEC), 117–22. Kuala Lumpur, Malaysia: IEEE, 2015. https://doi.org/10.1109/MySEC.2015.7475206.