iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** II **Month of publication:** February 2026

DOI: https://doi.org/10.22214/ijraset.2026.77579

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue II Feb 2026- Available at www.ijraset.com

Cipher Cache: Intelligent Secure Data Caching Framework for High-Performance Applications

Aditya Raj[1], Kavya Sinha[2], Sakshi Shailendra Bhavikatti[3], Aditya Kumar Chaurasia[4], Mrs. Aparna M[5]
Dept. of CSE, Dayananda Sagar College of Engineering, Bangalore, India

Abstract: ***The digital data fields like healthcare, finance, and cloud services are growing quickly, so it's important to have fast and safe ways to get data. The caching process improves performance by keeping often-used information saved, but many current systems don't do enough to keep that information safe, especially when it comes to using smart predictions. This paper looks at how intelligent and secure caching methods have developed over time, along with the challenges and drawbacks they face. Cipher Cache is a system that uses multiple layers of caching, mixes encryption methods, and uses machine learning to predict needs, so users can quickly and safely access data.***

Keywords - Edge Caching, Hybrid Encryption, Predictive Machine Learning, Multi-Layered Cache Architecture

I. INTRODUCTION

Continuously happening changes in the way industries use digital technologies have lead to huge rise in volume, speed and type of data that modern systems create and handle.. Different fields such as healthcare, finance, online shopping and smart city systems have started relying more and more on providing faster access to data for their important services. Caching has become important in such areas for speeding up the data retrieval by keeping often asked information nearer to the user. As the role of caching is working of such systems are extremely important, it is equally important to keep the cached data safe and private which has proved to be a problem in recent times.

Several traditional ways of caching such as LRU(Least Recently Used) and LFU(Least Frequently Used) are still used widely because they are easy to understand and make system faster without increasing the workload on main servers. These methods only fill up the cache after a request has been made by someone and data is removed based on how often it was used. This kind of reaction leads to poor use of cache especially in cases where how users act and popular content or data changes frequently. Relevant steps are not taken by many caching systems to protect sensitive data, often saving information in plain text. Organizations can be put at risk if cache servers are attacked. Researchers have made the combination of fast performance, strong security etc as their major area of interest. People are looking for systems can get data transmitted quickly and safely with also guessing what needs to be added next. The development of secure and smart caching over time has been looked upon by this review bringing together key findings and highlighting the areas of improvements from recent studies leading to the creation of CipherCache. It uses smart ML algorithms to predict needs, has multiple layers for storing data quickly and uses hybrid encryption techniques for security. A new standard for secure and fast storage can be set with the help of this where computers work together.

II. BACKGROUND AND RELATED WORK

A. Security and Privacy in Caching

Caching related early studies mostly revolved around collect things faster but didn’t spare much attention to keep the collected information safe. As modern application started collecting and dealing with private data, keeping cached data safe became more important. Gabry et al. (2016), researchers looked into secrecy constrained caching which means arranging cache in a way which keeps it secure as well as efficient. Similarly Xia et al. (2020), a method was suggested that focuses on security during caching in networks. Mathematical models were used in this approach to assess aspects of security when deciding where to place the cached data. Hybrid encryption has also been studied as a way to keep information safe without affecting the performance. Jaspin et al. (2021), it was shown that using both AES and RSA together works well for sending files over to the cloud proving that security can be improved without greatly slowing things down. Ni et al. (2021) looked more closely at mobile edge caching and found dangers like cache poisoning and inference attacks. Usage of secure key distribution was proposed as way to handle threats. Many of these approaches still see security as something added on and not built into the systems to predict and adjust how data is stored or retrieved even after improvements. This problem is solved by CipherCache by combining encryption with smart caching system retaining both safety and quick accessibility of data.

B. Privacy-Preserving Edge Caching

More privacy issues are created by edge caching since user data is kept nearer to the network's end points. In 2023, probabilistic caching strategies were introduced by Hassanpour et al. These strategies are used to randomly place content in cache to hide how often certain items are accessed. Sensitive information is helped to be prevented from being accidentally exposed. At the same time, the performance of the system is kept unchanged. In 2022, distributed reinforcement learning was used by Liu et al. to adjust caching in a way that privacy is kept.

Although these methods are found to work well at hiding who is accessing what, the system is usually made slower, and full encryption for the data stored in the cache is not provided. These techniques are improved by CipherCache through the use of predictive caching along with hybrid encryption, and good security and fast content delivery are provided.

C. Intelligent and Predictive Caching

Adaptability and performance of caching is achieved with the help of Machine learning. ICE which stands for Intelligent Caching at Edge developed in 2021 by Wang et al. uses deep reinforcement learning to improve how often data is cached and saves us the energy. Reinforcement learning was also used in mobile edge caching in 2018 by Xiao et al. to automatically adjust to interferences and changes in condition of network. These techniques increase efficiency but on the other hand overlook security and privacy. CipherCache addresses this issue by combining smart prediction with strong encryption, making it a caching system that can change and stay safe in ever changing network environment.

D. Hybrid Encryption in Caching and Cloud Systems

A combination of both symmetric and asymmetric encryption methods called Hybrid encryption has been shown to work well in keeping cloud data safe (Jaspin at al., 2021). Its use in caching system has not been very widespread. A mix of encryption methods in caching system is used by CipherCache to keep data safe in both cases when it’s stagnant and while it’s sent across. A major weakness in how traditional caching systems work and keep important information is fixed using this method.

E. Broader Surveys and Benchmarking

Comprehensive surveys were conducted by Zhang et al. (2025) and Li et al. (2024), where it was highlighted that very few frameworks are successfully combined with predictive intelligence, privacy, and security in a single system. While strategies are categorized and trade offs between performance and security are assessed in previous studies, integrated implementations are still found to be a minimum. This gap is addressed by CipherCache, where predictive learning, privacy-preserving measures, and hybrid encryption are combined into a single framework and both high performance and secure caching are achieved.

Even though some major advances have been made in caching, security and machine learning big problems are still faced by current systems. A major issue is that the efforts made to improve speed and safety are not working well together. Many existing systems are either using smart predictions to make things faster or using encryption and privacy tools to protect data but these features are not usually combined into one clear and organized system. This separation causes solutions to perform well in one area but fail in another making them not enough for today’s demanding applications.

Another issue is that too much is relied on rules that act only after something has happened. Traditional caching methods like Least Recently Used (LRU) or Least Frequently Used (LFU) are focused on past data requests instead of predicting what might be needed next. This way of working can cause more cache misses waste resources and slow down performance especially when user behavior and data demand keep changing. Security is also a major problem in many caching systems.

Cached data is often kept in plain text so it can be easily seen or stolen if someone gets access to the cache server. This is very risky in fields like healthcare and finance where important data must be kept safe from unauthorized access. Some systems also lack real time visibility and cannot manage scaling well in large environments. Without proper monitoring tools or ways to fix problems fast the system becomes less reliable and harder to maintain. Even though privacy preserving and learning based methods are strong they need a lot of computing power and resources. Some methods like distributed reinforcement learning or probabilistic caching can work well but may fail when resources are low making them hard to use in real cases.

These problems show that caching systems are needed which can balance speed safety and resource use without becoming too costly. CipherCache is made to solve these problems by giving one system that uses many caching layers machine learning to predict needs and mixed encryption techniques. This setup makes sure data is delivered fast safely and smoothly and can scale well

in different environments. CipherCache is a major improvement because it combines smart features strong security and flexibility in one system fixing the issues faced by existing caching tools.

III. OUR APPROACH

CipherCache is a modern and smart system for storing data securely in a fast and efficient way to meet the needs of today's high performance applications. Its design is based on a few main ideas that work together to make sure data is accessed quickly dependably and safely.

At the core of CipherCache the use of predictive machine learning models is done. Smart methods like LSTM, ARIMA and reinforcement learning are used by the system to predict how users will access data and to prepare the needed information early. By predicting what content will be in high demand cache misses are reduced and delays are lowered so popular information is always ready when needed.

A multi layer cache setup is also used in the system where in memory caches like Redis and Memcached are combined for fast access along with distributed cache groups such as Hazelcast and Apache Ignite to support scaling and handle failures. A good balance between fast performance and strong reliability is found through this layered setup and smooth service is ensured even when workloads change or some parts fail.

A mix of encryption methods is used by CipherCache to protect cached data. AES is used for quick data encryption and RSA is used for safe key sharing and handling. Through this two layer encryption method important data is kept secret and safe from changes whether it is stored in the cache or moving through the network.

A real time dashboard is also included in CipherCache where detailed information about cache performance can be seen such as hit and miss rates data retrieval times extra delay caused by encryption and the status of each server node. This visibility helps system adjustments to be made before issues occur and possible problems or slowdowns to be found quickly which improves the overall working of the system.

A framework to be work well in different areas such as healthcare, finance, IoT etc is built to be strong and able to grow. The growth and ease to work with current systems is allowed by its design keeping up with changing needs without affection its working and safety. CipherCache overcomes the shortcomings of current systems by combining smart predictions, strong security measures and efficient data storage methods.

IV. IMPLEMENTATION

Python with Flask is implemented as the primary interface framework in the system. The system also integrated machine learning based predictive caching with hybrid cryptography (AES+ RSA). The system workflow is structured as below:

1) Input Module (Dataset Loader): This module is responsible for loading domain specific datasets that contain historical user request patterns. A unique schema can be found with each dataset. Custom dataset integration can be supported by loader for seamless expansion with making any code changes.

2) Predictor Engine: Analysis of past request patterns for identification of most likely future data requests is handled by this model. Generalized patterns can be learnt from all datasets enabling the system to facilitate domain agnostic predictive caching.

3) Secure Cache Manager: Handling caching operations along with ensuring complete data security is the responsibility of this module using a hybrid encryption model. The various tasks performed include:

- Encrypting data using AES technique before caching
- Applying a RSA wrapping over the AES keys to secure them
- Decrypting data at retrieval
- Cache hits, misses as well as expiration policies management

4) Redis Caching Engine: Since Redis serves as the primary in memory caching layer in our system, it is responsible for:

- Storage of the data that was encrypted
- Facilitating faster read/write operations
- Handling preloaded predictions sent from the SCM (Secure Cache Manager)

5) User Interaction Interface (Flask Web UI): The frontend of the system built using Flask provides simple interface to interact with the said system. At this interface users can:

- Enter manual search or queries

- View real time cache hits and misses
- Monitor predicted and preloaded items
- Access the dashboard

6) Performance Monitoring Dashboard: A Grafana dashboard helps display live analytics including cache miss vs hit ratios, response latency, encryption decryption overhead, successful preloading of predicted items.

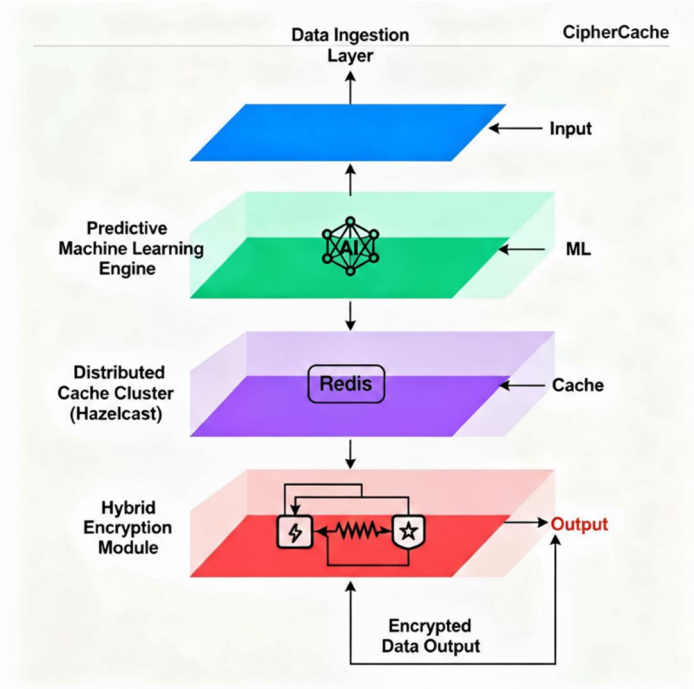


Fig. 1. Overall system workflow of CipherCache

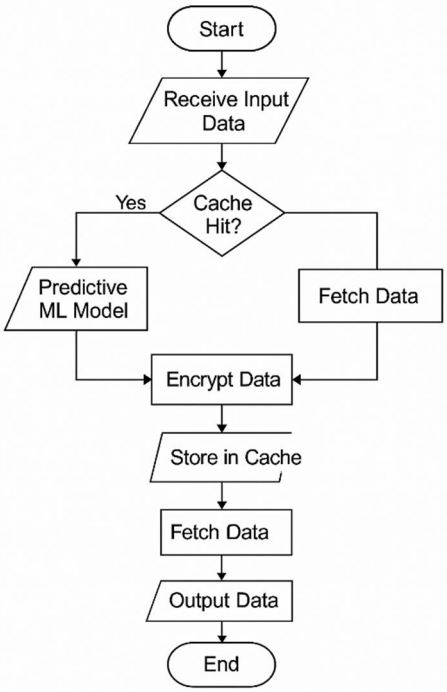


Fig. 2. Architectural flow of CipherCache

V. RESULTS AND ANALYSIS

To validate the performance of the system, an extensive evaluation was conducted using three diverse sample datasets representing healthcare, banking and e-commerce request patterns. To assess the system's predictive accuracy, encryption and decryption overhead, latency reduction and caching efficiency, a total of 14,000 request entries were used from the mentioned datasets.

The evaluation of the model was carried out in two phases. The first stage involved the machine learning based predictor which was used to test predicted accuracy and hit rate improvement across several domains.

The second stage involving the SCM (Secure Cache Manager) was evaluated for its capacity to encrypt, store and retrieve data with minimal overhead ensuring a perfect balance is maintained and the performance gain is not traded with security based delays.

When compared to traditional caching policies such as LR u and LFU, CipherCache provides improvement in cache hit rates and data retrieval latency by 40-60 % and 15-30% respectively. As the system is domain agnostic, it performs uniformly well across healthcare, banking and e-commerce datasets without domain specific tuning.

There is an introduction of only minimal overhead by hybrid encryption (AES+RSA) validating its efficiency. The results also demonstrate that the predictive engine consistently identified items with strong accuracy enabling proactive preloading.

VI. CONCLUSION AND FUTURE WORK

Simple designs of caching systems that focused on speed had an evolution to more advanced ones that also care about safety, keeping information private and making smart decisions. Speeding up data access and lowering delays were major reasons behind designing of caching solutions in initial days. With the increasing complexity of digital systems and need to handle more data , caching methods have to become safer and smarter as well. Caching tools have become more than just how fast they are for people as they also care about the how well these tools can keep their data safe and make smart decisions while working with data in changing and spread out systems. Creating a single system that offers fast performance, better security and smart prediction is still a difficult task even after many improvements and advancements.

A natural balance is struck by many current solutions, focusing on speed make the data less secure and using strong encryption can slow things down and reduce the amount of data processed at once. Usage in fields such as finance, healthcare etc is difficult under these rules.

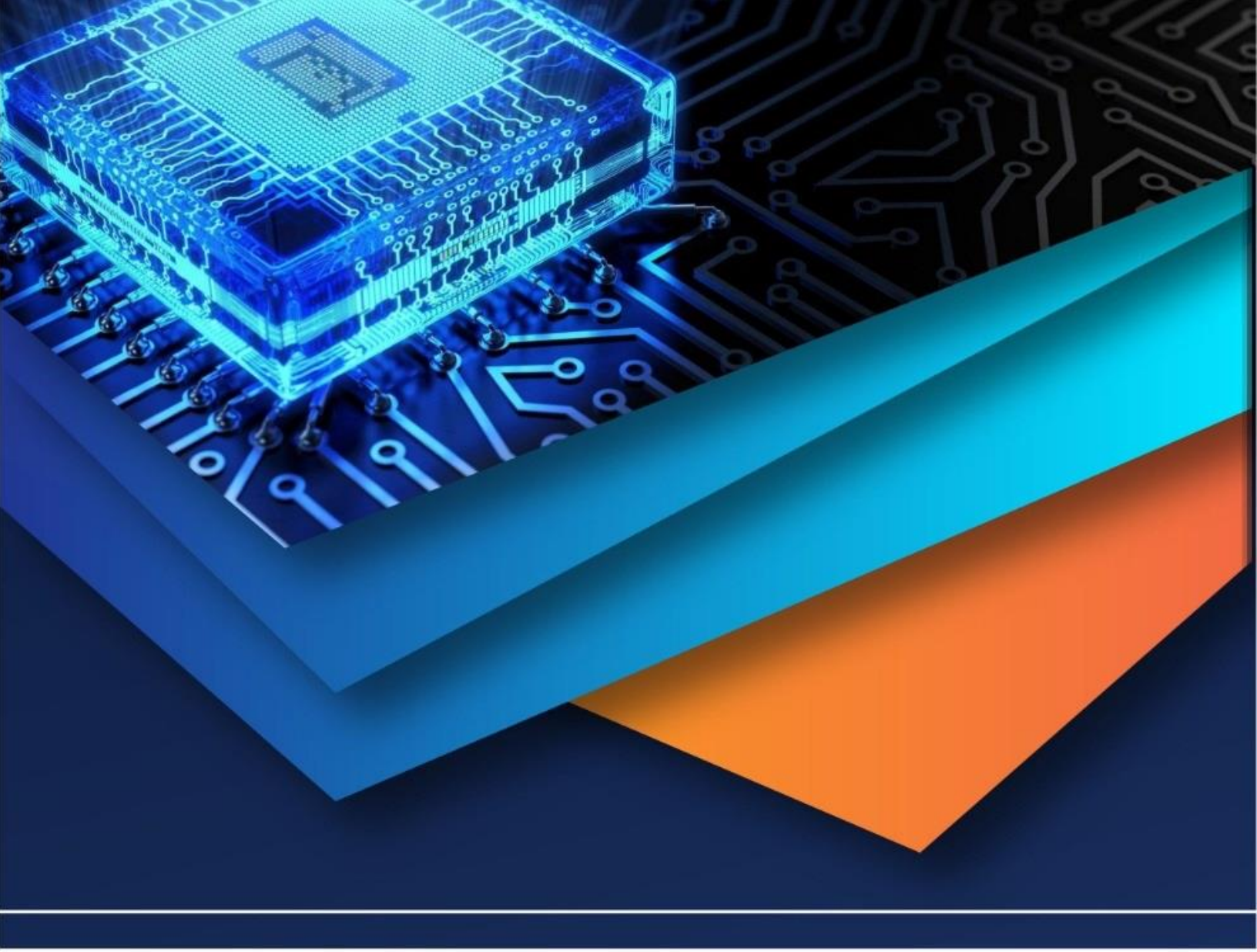
CipherCache proves to be a major improvement in this situation. ML based prediction, multiple levels of caching and mixed encryption methods are combined in a single design. Data is kept secure and accurate along with quick access due to this mixed framework.

A new way to build safe, fast and smart caching systems by combining strong security with better performance is offered by CipherCache. Features like federated learning to train model while keeping user's data private while training models, using blockchains to make logs unchangeable and easier to check can be part of future improvements depending upon the circumstances. CipherCache will be very important in building strong and safe digital systems as data becomes bigger and more important.

For future expansion of this project, powerful predictors like LSTMs or Transformers can be used for improvement of accuracy. Cloud Deployment on Amazon Web Services (AWS) or Google Cloud Platofrm (GCP) can handle real world testing.

REFERENCES

[1] Gabry et al., "On Edge Caching with Secrecy Constraints," 2016.
[2] Xia et al., "Security-Aware Caching Placement Optimization in Cooperative Networks," 2020.
[3] Jaspin et al., "Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA," 2021.
[4] Ni et al., "Security and Privacy for Mobile Edge Caching: Challenges and Solutions," 2021.
[5] Hassanpour et al., "Privacy-Preserving Edge Caching: A Probabilistic Approach," 2023.
[6] Liu et al., "Distributed RL for Privacy-Preserving Dynamic Edge Caching," 2022.
[7] Wang et al., "ICE: Intelligent Caching at the Edge," 2021.
[8] Zhang et al., "A Survey on Privacy-Preserving Caching at Network Edge," 2025.
[9] Li et al., "A Survey of Edge Caching: Key Issues and Challenges," 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY