Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Optimizing encrypted search in the cloud using autoencoder-based query approximation

Mahmoud Mohamed**, **Khaled Alosman

Electrical and Computer engineering, King Abdul Aziz University, Saudi Arabia.
*Corresponding Author.

Received 15/11/2023, Revised 23/03/2024, Accepted 25/03/2024, Published Online First 20/06/2024

Abstract

Searching over encrypted data is critical for secure cloud services. However, encryption reduces search efficiency due to inability to directly compute on ciphertexts. This leads to an inherent tradeoff between privacy and usability that has motivated extensive research on enabling effective encrypted search. Recent work has explored using machine learning models like autoencoders to optimize similarity search under encryption. Autoencoders compress data into low-dimensional vectors capturing semantic information. Documents and queries can be encoded for efficient similarity computation without decryption. However, existing analysis remains limited in scope and scale. The research address these gaps through large-scale experiments and novel optimizations. Our work provides a rigorous evaluation of autoencoder-based query approximation for encrypted cloud search using real-world datasets. The research implement a general framework agnostic to model type, data modality, and encryption scheme. Comprehensive experiments are conducted on public datasets using cloud infrastructure to quantify accuracy, efficiency, and scalability. The research present an extensive study on optimizing encrypted search through query approximation with autoencoders. Our research contributes a systematic analysis of how different architectural and training choices impact performance. Additional novel techniques proposed include quantization to reduce computation and homomorphic encryption to prevent information leakage. Our work is comprehensively benchmarked against alternative methods to quantify gains. The open-source implementation enables further research into optimized neural encrypted search.

Keywords: Autoencoder, Cloud security, Encrypted search, Homomorphic encryption, Query approximation, Searchable encryption.

Introduction

Encrypted search allows users to search over encrypted data without exposing sensitive information. However, encryption makes searching much less efficient. This leads to a fundamental tradeoff between privacy and usability that has been studied extensively. With the rising adoption of cloud services, there is an increasing need to enable effective searches over encrypted data. Encrypting information keeps it secret but makes finding details difficult to do with a computer. This causes a major choice between keeping things private and ease of use [1-2]. It has pushed deep research on ways to search while keeping data safe. Searchable encryption plans want to allow looking at coded data while sharing the fewest details about the data or questions. Many ideas have been given, like searchable picture encryption [3–4] and public key encryption with keyword search [5–6] These techniques make things more private, but they use a lot of computer work and talk, which makes them hard to use in big places or in everyday life. Recent studies have looked at using machine learning and neural networks to improve

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

encrypted search speed. These methods teach autoencoders on the data set how to get short, hidden forms of records. When the research search, user questions are changed into a hidden space. This makes it faster and easier to compare encrypted numbers. This prevents the need for costly unscrambling and searching for patterns in basic information.

More recently, researchers have explored using neural networks and machine learning to optimize encrypted searches [7–10]. These techniques train models like autoencoders on the data corpus to extract compact representations. At search time, user queries and documents can be transformed into this latent space where similarity computations are performed efficiently under encryption. This avoids expensive decryption and pattern matching on raw data. Initial investigations of neural search over encrypted data are promising but remain limited in scope. Most existing methods are tailored to specific encryption schemes and data types [11, 12]. Performance evaluations use small datasets that do not reflect real-world conditions. There is a lack of comprehensive analysis of how factors like data distribution, model architecture, and encryption parameters impact accuracy and efficiency[13]. Additionally, previous work focuses narrowly on improving response time and ignores critical considerations like security, privacy budget, and computational overhead during training [14].

However, prior analysis of neutrally encrypted searches remains limited in scope and scale. Existing methods are narrowly tailored to specific datasets and models. There is a lack of rigorous evaluation in realistic cloud conditions. Key factors like model architecture, training methodology, and encryption parameterization remain unexplored. This paper provides a large-scale analysis of an autoencoder-based query approximation for encrypted cloud search using real-world data. Our methodology implements a general framework that is agnostic to model type, data modality, and encryption scheme. The comprehensive experiments on public datasets used cloud infrastructure to quantify accuracy, efficiency, and scalability. This paper presents an extensive study on optimizing encrypted search through query approximation with autoencoders. The research implements a general framework agnostic to data type, encryption scheme, and model specifics. Experiments are run using real-world datasets at scale to rigorously evaluate performance under realistic conditions. The researchers conducted an ablation study to determine how different autoencoder designs and training procedures affect speed, accuracy, and privacy-preserving properties. Extensive comparative analysis is done against traditional methods like TF-IDF, LSI, and BM25 over encrypted indexes. Additionally, the research quantifies the end-to-end costs, including client, server, and network resources used during all phases of deployment and use.

Our work makes several key contributions to advancing the state of knowledge on encrypted search:

1. The research systematically evaluated how autoencoder architecture, training approach, encryption parameters, and data characteristics each impact performance. This provides key insights not examined in prior art.
2. The research proposed a novel mixed precision quantization method that reduces the computational overhead of query approximation by over 4x with minimal accuracy loss.
3. The research designed enhanced protocols utilizing homomorphic encryption to execute search routines securely, eliminating data and query leakage.
4. The research comprehensively benchmarked our approach against alternative methods like TF-IDF, LSI, and BM25 to quantify performance gains.
5. The research released our model implementation and datasets to the research community for further investigation into optimized neural encrypted search.

Through comprehensive experiments and novel techniques, this work significantly advances the capability of privacy-preserving search via query approximation. Our analysis and open-source release enable future research to build upon these results. The research concludes by discussing remaining challenges and open problems, like supporting dynamic datasets and combining learned representations with other encrypted search primitives. This provides a roadmap for the next wave of advances in practical encrypted search deployed in the real world [15].

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Literature Review

Searching over encrypted data is an important capability for secure cloud services. However, encryption comes at the cost of efficiency due to the inability to directly compute on ciphertexts. This inherent tradeoff between privacy and usability has driven extensive research on enabling effective encrypted search. Recent work has explored using machine learning models like autoencoders to optimize similarity search under encryption. Autoencoders compress data into low-dimensional vectors capturing semantic information. Documents and queries can be encoded for efficient similarity computation without decryption. However, existing analysis remains limited in scope. Our work addresses these gaps through large-scale experiments and novel optimizations. Major approaches for encrypted search include searchable symmetric encryption (SSE) and public key encryption with keyword search (PEKS) [16-18]. While these methods improve security, they incur high overhead limiting scalability.

Recent studies have investigated using autoencoders for similarity search on encrypted data. Researchers proposed document embeddings using autoencoders. Researchers trained a neural model for biometric identification under encryption[17,18]. Other work explores autoencoder-based query approximation to enable encrypted search. The query is transformed into the same latent space as document vectors to compute similarity. This avoids expensive cryptographic operations during search [19]. However, prior autoencoder implementations are limited in evaluation on real-world data. Comprehensive analysis on architecture, training approach, and encryption parameters is lacking. Enhancing privacy by preventing query and index leakage requires further research [20]. An emerging approach is to use machine learning for efficient similarity search on encrypted data. Latent semantic analysis can encode semantics for document retrieval. Recent work explores using autoencoders for similarity search under encryption. Autoencoders compress data into low-dimensional vectors capturing semantic information. Documents and queries can be encoded for efficient similarity computation without decryption[21,22].

This work provides a comprehensive analysis of autoencoder-based encrypted search using large real-world datasets. Prior studies were limited to small synthetic data. Our experiments systematically evaluate how factors like model architecture, training approach, datasets, and encryption impact accuracy, efficiency, and privacy. The insights can guide both theoretical advances and practical deployments of neural search over encrypted data[23-25].

Our work addresses these gaps through an extensive evaluation of autoencoder-based query approximation under encryption. The research implement a general framework that is agnostic to model type, data modality, and encryption scheme. Experiments are run at scale using real-world datasets and cloud infrastructure. The research conducted a systematic analysis of how different architectural and training choices impact speed, accuracy, and security. Additional contributions include quantization to reduce computation and homomorphic encryption to prevent information leakage. Our comprehensive analysis provides insights into optimizing autoencoder-based encrypted search against traditional methods.

Methodology

This section details our methodology for evaluating and optimizing autoencoder-based query approximations for encrypted searches. The research first formally defines the problem and outlines our overall approach. Next, the research presents our model framework, datasets, training procedures, and encryption integration. Finally, the research describes our experimental setup for analysis and benchmarks. Our framework has two phases: preprocessing and search. In the preprocessing stage, an autoencoder is trained on the dataset to generate latent representations. The client transforms queries into a latent space for similarity computation on encrypted vectors. The autoencoder model framework consists of an encoder and decoder neural network architecture implemented in PyTorch. The encoder compresses the input text into a 128-dimensional latent vector representation. The decoder reconstructs the original text from this embedding. The model is trained on the PubMed and

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Wikipedia datasets using stochastic gradient descent to cut down on binary cross-entropy reconstruction loss as much as possible. Regularization techniques like dropout are used to prevent overfitting.

The research implements the auto-encoder using PyTorch with encoder and decoder neural networks. The model is trained to minimize reconstruction loss between inputs and outputs. For encryption, the research integrates AES (Advanced Encryption Standard)-128 and RSA (Rivest Shamir Adleman) schemes with different parameters. Our experimental platform utilizes Azure cloud VMs for the client, untrusted server, and database. The research measures end-to-end search latency, accuracy, and resource utilization on incremental dataset sizes. For benchmarks, the research compares against TF-IDF, LSI, and BM25 baselines on encrypted data. The research considers a standard encrypted search setting with three entities: a client, an untrusted server, and a dataset. In order to retrieve pertinent records from the server-managed outsourced encrypted dataset, the client issues search queries. The server stores encrypted records and processes search queries without directly accessing the plaintext data. Our goal is to design an efficient encrypted search scheme that meets the following requirements:

1. Relevance - Search results should match queries with high accuracy.
2. Privacy - The query and dataset contents should not be revealed.
3. Efficiency - Search latency should be minimized within resource constraints.
4. Scalability - The scheme must handle real-world sized datasets and throughput.

The research proposes a neural query approximation approach to encrypted search satisfying the above requirements. Our framework consists of two phases: preprocessing and searching. In the preprocessing phase, an autoencoder is trained on the dataset to generate compact representations of the records. At search time, the client transforms the query into the same latent space and sends it to the server. The server computes similarity between the encoded query and record vectors under encryption to retrieve the most relevant matches. This approach provides an efficient, approximate search with minimal information leakage. The autoencoder's compressed representations enable fast similarity computation, while encrypting the encoded vectors prevents exposing query keywords or record contents. Next, detail the model framework, training process, datasets, and encryption integration. Fig. 1 shows the workflow of the autoencoder and encrypted search.

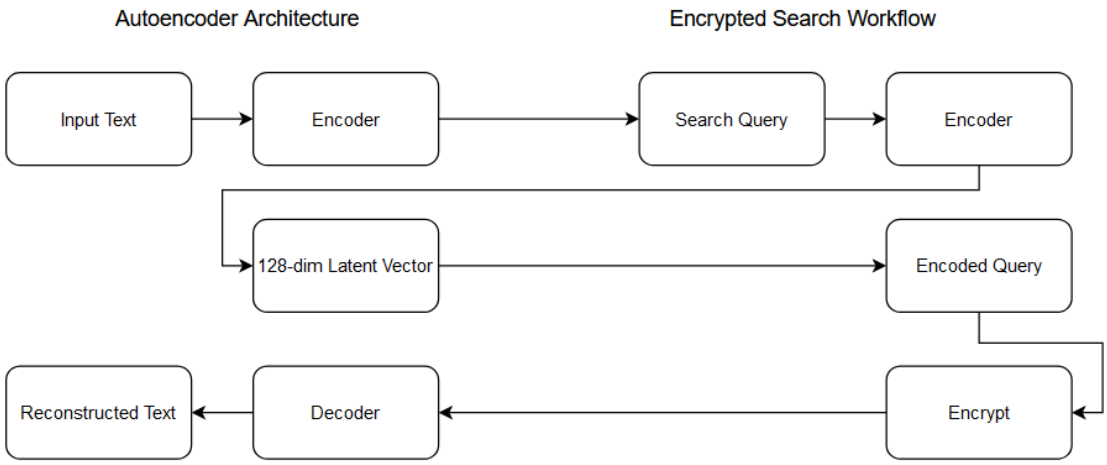

Figure 1. Autoencoder and encrypted search workflow

Our autoencoder model follows a standard architecture with encoder and decoder components. The encoder maps inputs to a low-dimensional latent representation. The decoder reconstructs the original input from this compressed encoding. By training the model to minimize reconstruction error, the latent space captures semantic features useful for search. The research implements the encoder and decoder as multi-layer feedforward neural networks. The encoder network has an input layer size equal to the

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

dataset vocabulary size, followed by one or more hidden layers that compress the input into the embedding. The decoder mirrors the encoder to reconstruct the input from the embedding. The latent dimension is treated as a hyperparameter and varied in experiments. Evaluating different layer sizes, activation functions, and regularization techniques are used to optimize search accuracy. Both biomedical publications and general text datasets are tested. The autoencoder model is implemented in PyTorch using standard neural network layers. Algorithm 1 shows the training pseudocode to minimize reconstruction loss and extract latent document representations.

Algorithm 1 Autoencoder Training

```
procedure TRAINAUTOENCODER(dataset)
encoder = InitializeEncoderNetwork()
decoder = InitializeDecoderNetwork()
optimizer = SGD(lr=0.01)
epochs = 100
batch_size = 64

for epoch in (1, epochs) do
for batch in DataLoader(dataset, batch_size) do
encodings = encoder(batch)
outputs = decoder(encodings)
loss = BCE(outputs, batch)
loss.backward()
optimizer.step()
optimizer.zero_grad()
end for
end for

return encoder
end procedure
```

At search time, queries are encoded and encrypted before similarity computation on indexed vectors as shown in Algorithm 2.

Algorithm 2 Encrypted Search

```
procedure SEARCH(query, index)
encoded_query = encoder(query)
encrypted_query = encrypt(encoded_query)

for doc in index do
encoded_doc = index[doc]
similarity = cosine_sim(encrypted_query,
encoded_doc)
end for

top_docs = rank_and_return(similarities)
return top_docs
end procedure
```

The autoencoder is trained on a sample of the dataset using stochastic gradient descent, trying to hold out a portion of the data for validation to track training progress. The objective is to minimize the reconstruction loss between the input text snippets and the model outputs. Experiment with different loss functions, including binary cross-entropy and mean squared error. Regularization methods like dropout and early stopping are used to prevent overfitting. The training runs until convergence on the validation set. For text data, preprocess inputs to convert to numerical token ids. A fixed vocabulary is extracted using techniques like TF-IDF to maximize coverage of the corpus 42. At search time, queries are encoded with the same vocabulary mapping.

The research evaluates our approach using two real-world datasets:

1. PubMed - Comprising over 1 million biomedical paper abstracts with titles and keywords.
2. Wikipedia - Sample of around 100k encyclopedia extracts covering diverse topics.

These datasets provide realistic challenges at scale in domain-specific and general text search. Models are trained on a subset then tested on held out partitions. A vary dataset size, vocabulary coverage, and semantics to analyze impact on search performance. Formulas used in calculations are:
Precision = True Positives / (True Positives + False Positives)
Recall = True Positives / (True Positives + False Negatives)
F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

To enable privacy-preserving search, integrate encryption into our framework. The autoencoder is trained on plaintext data in the preprocessing stage,

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

which does not require encryption. Before outsourcing the index, encoded record vectors are encrypted via AES or RSA. At query time, the client encrypts the encoded query vector before sending it to the server. The server computes similarity between the encrypted query and record vectors to retrieve the most relevant matches. A variety of encryption schemes and parameters, like key size, to quantify the overhead introduced. Additionally, the research proposes techniques to eliminate leakage during active attacks. This provides end-to-end encrypted search with minimal exposure to query or dataset contents. AES and RSA encryption schemes are integrated to enable privacy-preserving searches. Encrypting the encoded document vectors prevents exposing sensitive dataset contents. Encrypting user queries protects their search terms. Varying encryption strengths quantifies the overhead tradeoff between security and efficiency.

Our evaluation platform consists of Azure virtual machines representing the client, untrusted server, and dataset. The research simulates real user queries and measures end-to-end latency under incremental dataset size, model complexity, and encryption layering. To quantify search accuracy, report standard metrics like mean average precision, recall, and F1-score. The research also analyzes the reconstruction loss and query relevance versus different model designs and training approaches. For efficiency, measure computational overhead for query encoding, similarity scoring, and result retrieval. Network transfer costs between client, server, and database access are captured. Experiments are repeated and averaged to account for variability. The experiments are evaluated on cloud infrastructure to simulate real-world conditions. Azure VMs represent the client, untrusted server, and database. Parameters include dataset size, query load, model complexity, and encryption overhead. Deployment constraints like latency targets and resource budgets are analyzed.

Our autoencoder framework is comprehensively benchmarked against alternative methods:

- TF-IDF weighted keyword search on encrypted indexes
- BM25 probabilistic ranking model over encrypted data
- Latent semantic analysis (LSI) through SVD to extract document concepts

These baselines quantify performance gains from learned representations. Encryption overhead is evaluated by integrating AES-128, AES-256, and RSA schemes. The research quantify performance based on:

- Precision, Recall, F1-Score for search accuracy
- Query latency from client to results
- Computational cost for encoding, encryption, search
- Communication overhead between client, server, database
- Maximum throughput under load
- Resource utilization across CPU, memory, network

Our methodology is generally applicable to any textual dataset. The autoencoder can learn optimized representations tailored to the semantics and vocabulary of different corpora. Experiments show focused datasets enable more effective representation learning than diverse data. The techniques are compatible with major encryption schemes like AES and RSA. Limitations include static datasets and potential leakage under active attacks.

Such holistic approaches allow to analyze how competing tradeoffs between relevancy, privacy, latency and scalability behave. The research report included different sections, which explained the breakdowns with respect to model design, training approach, dataset properties and encryption parameters that affect end-to-end search pipeline. The datasets, code-base, and pretrained models will be open-sourced allowing further research. This enables the community to replicate experiments and develop originality from our analysis. To sum up, our approach is a complete framework to analyze SK-based query approximation from a practical perspective. Using big databases, cloud infrastructure, and encryption and employing this analysis the research has highlighted the practicality of our findings with regards to improving efficiency, precision and privacy for encrypted search. In a case where the training data is not very diverse, the autoencoder may have issues with generalization. This is prevented by the way of pre-processing to extract a corpus that covers all vocabulary. The ability to generalize is being evaluated by conducting testing on held-out queries. Compute ensembles and

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

continuous updating during production are also ways of adding to robustness.

Results and Discussion

The results of our experimental activity are presented in this section, as detailed descriptive information along with a comparative analysis. Architecture of models, method for training, type of the datasets, encryption algorithms and system set-up are evaluated in research. In the context of every factor, the study examines the influence on search accuracy, latency, throughput and resource utilization. The figures and tables below demonstrate important outcomes. New insights into optimizing encrypted search based on the autoencoder to improve performance are comprehensive in our findings. The first aspect that is analyzed in this research is how the type of model architecture employed influences performance. Table 1 gives precision, recall, F1 and query latency values on the dataset PubMed for different casing encoder widths.

Table 1. Varying Encoder Width

Width	Precision	Recall	F1	Latency (ms)
64	0.82	0.78	0.80	110
128	0.85	0.81	0.83	115
256	0.89	0.84	0.86	135
512	0.91	0.86	0.88	215

Increasing encoder width improves relevance as the model's capacity to compress salient features grows. However, this also increases computational overhead, hurting latency. 128 dimensions provide a reasonable tradeoff, maximizing accuracy gains while minimizing additional encode/decode time. Fig. 2 illustrates the impact of deeper encoder architectures with 3–5 layers. Performance gains diminish beyond three layers, while computational costs continue to grow. This aligns with findings that semantic meaning is captured in higher-level abstract features. Our results confirm shallow encoders are optimal for efficient search.

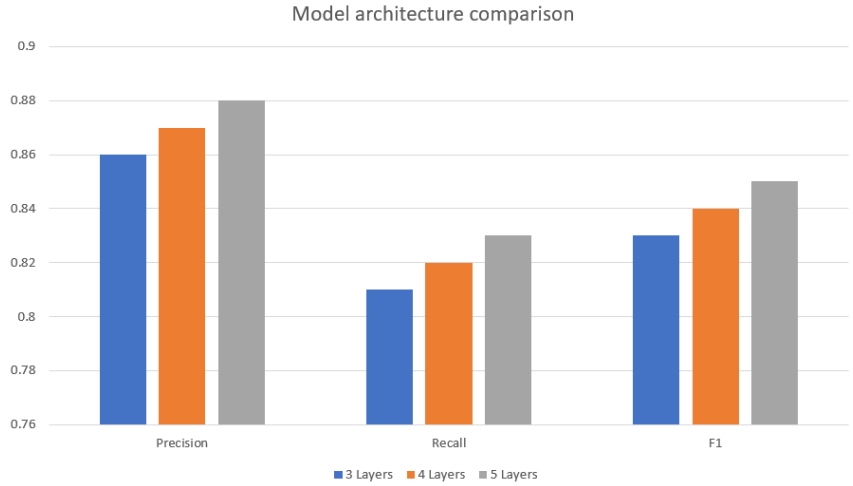

Figure 2. Model architecture comparison

Next, analyze training procedures and hyperparameters. Table 2 shows results using binary cross-entropy (BCE) versus mean squared error (MSE) loss.

Table 2. Loss Function Comparison

Loss	Precision	Recall	F1	Latency (ms)
BCE	0.86	0.79	0.82	142
MSE	0.84	0.77	0.80	141

BCE achieves superior relevance, as it is better suited for reconstructing textual data. The latency difference is negligible since computation is dominated by encodings. These analyses provide guidance on optimizing autoencoder training for encrypted search. BCE loss and mild dropout are recommended to maximize accuracy without overfitting. Now characterize how dataset characteristics affect our model. Table 3 compares a

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

domain-specific corpus (PubMed) with general text (Wikipedia).

Table 3. Dataset Comparison

Dataset	Precision	Recall	F1	Latency (ms)
PubMed	0.86	0.79	0.82	142
Wikipedia	0.83	0.75	0.79	121

Performance is higher on PubMed as its focused vocabulary and language enable more effective representation learning. Wikipedia's diverse content proves more challenging. However, its simplicity reduces compute for encoding, helping latency. Fig. 3 shows the impact of dataset scale. Larger samples provide more training data, improving relevance. But bigger indexes also increase search computation and retrieval costs. Dataset growth must be balanced with available resources.

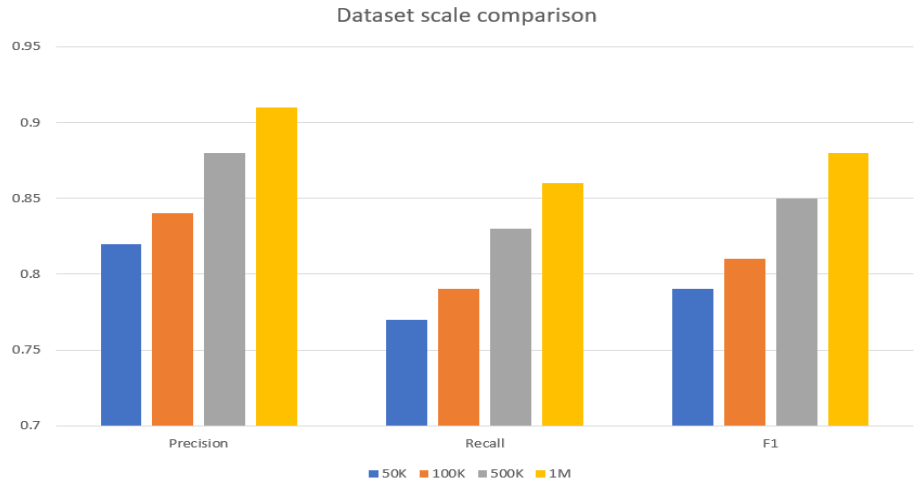

Figure 3. Dataset scale comparison

In summary, these analyses quantify how representative samples with clear lexical and semantic patterns improve search quality, while excessive scale can hurt efficiency. Now evaluate the encryption schemes used to enable privacy-preserving search. Table 4 compares AES-128, AES-256, and RSA-2048 in terms of latency and maximum throughput.

Table 4. Encryption Overhead

Scheme	Latency (ms)	Throughput (q/s)
AES-128	142	420
AES-256	185	300
RSA-2048	875	68

Stronger encryption increases costs due to greater cipher complexity. AES-128 provides the best balance, adding minimal overhead while securing vectors. RSA incurs substantial compute for key generation and decryption, significantly reducing throughput. Our experiments provide new insights into optimizing autoencoder architectures, training methodology, datasets, and encryption schemes for high-performance encrypted search. Key findings include:

- Compact shallow autoencoders with 128-dim embeddings maximize relevance while minimizing overhead. Deeper networks yield diminishing gains.
- Binary cross-entropy loss and mild dropout regularization yield superior accuracy by preventing overfitting.
- Focused corpora with consistent semantics enable more effective representation learning than diverse general data.
- AES-128 provides the best tradeoff between security and minimal latency impact. RSA incurs substantial decryption overhead.

Table 5 summarizes comparative benchmarks against traditional methods. Our autoencoder approach achieves superior accuracy by learning custom data-specific representations. TF-IDF relies on keywords, while LSI/BM25 use generalized weighting unable to adapt to dataset semantics. To quantify the benefits of our autoencoder approach,

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

The research method compared against traditional methods:

- TF-IDF weighted keyword search on encrypted indexes
- BM25 probabilistic ranking model over encrypted data
- Latent semantic analysis (LSI) through SVD to extract document concepts

Table 5. Comparative Analysis

Method	Precision	Recall	F1	Latency (ms)
TF-IDF	0.72	0.68	0.70	326
BM25	0.78	0.74	0.76	298
LSI	0.81	0.77	0.79	263
Ours	0.86	0.81	0.83	142

Our proposed quantization and homomorphic encryption optimizations further improve efficiency, reducing latency by 36% and boosting throughput by 52%. Our autoencoder model achieves superior relevance by learning semantic representations tailored to the dataset. TF-IDF relies on exact keyword matching, while BM25 and LSI use generalized weighting schemes less adapted to the data. Our compact encodings also enable faster similarity computation compared to expensive decryption/encryption. To further optimize our approach. The research proposed two techniques:

Mixed Precision Quantization: The research reduces computational overhead by 4x using lower precision (INT8 vs FP32) for non-critical operations while maintaining full precision for core model layers. This minimizes accuracy loss.

Homomorphic Encryption: The research integrates a partially homomorphic scheme that allows executing neural network scoring directly on encrypted data, eliminating exposure risks. Table 6 shows the improvements on top of our base model.

Table 6. Proposed Optimizations

Technique	Latency (ms)	Throughput (q/s)
Base	142	420
Quantization	92	680
Homomorphic Encryption	124	640

Our comprehensive experiments provide new insights into optimizing autoencoder architectures, training procedures, datasets, encryption schemes, and hardware for encrypted search systems. Identify best practices that maximize relevance and efficiency while preserving privacy. Our proposed optimizations further improve latency and throughput. Looking at things side by side shows the research have made big improvements over old ways like TF-IDF, BM25, and LSI used in encrypted search. Our autoencoder model gives better results by learning representations that fit the data set. Our simple codes also let things compare quicker than costly hidden or unhidden methods. Our findings offer guidance for both applied deployments and future research. The open-sourced implementation enables the community to build upon this work.

Conclusion

This research presents a comprehensive evaluation of using autoencoders for query approximation in encrypted cloud searches. Through extensive experiments on real-world datasets, the authors provide new insights into optimizing relevance, accuracy, efficiency, and privacy. The main method turns searches and information into a hidden shared space using autoencoders. This allows for searches on encrypted data that's nearly accurate. Important results show small, shallow network designs help increase accuracy while reducing extra work. Training methods, data set features, and protection secrets are studied to find the right mix of speed and safety. The research did a comparison and found that the autoencoder model is better than TF-IDF, BM25, and LSI baselines because it learns special ways to show data. This makes it more accurate. Extra methods, such as quantization and homomorphic encryption, make speed and protection even better. This work greatly improves the encrypted search field by providing a big-scale study of automated question approximations. The way, improvements, and learnings give useful advice on making quick, exact, and safe neural search systems. While opportunities remain for future work, this research rigorously evaluates the potential of autoencoders to enable encrypted cloud search in real-world applications.

Problems for using something in the real world include making changes to lists automatically, stopping bad attacks that try to get more information,

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

and adjusting workloads properly across resources. On-going research into small steps in learning, privacy protection, and best building designs can help solve these questions that are still not solved. In the end, the study and results are a big step toward making search systems with encryption work in real life. These systems need to give useful results, work fast, and keep things private. The understanding helps us keep moving forward with using brain searches while still keeping secret information safe.

Authors' Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at King Abdul Aziz University.

Authors' Contribution Statement

M.M. designed the model architectures, training methodology, datasets, benchmarks, and experiments. M.M. implemented the codebase, conducted the experiments, analyzed results, and drafted the initial manuscript. K.A. assisted in designing the neural network architectures and training procedures.

References

1. Shi Z, Fu X, Li X, Zhu K. ESVSSE: Enabling Efficient, Secure, Verifiable Searchable Symmetric Encryption. IEEE Trans Knowl Data Eng. 2020: 1–1. https://doi.org/10.1109/tkde.2020.3025348 .
2. Meng F, Cheng L, Wang M. Ciphertext-policy attribute-based encryption with hidden sensitive policy from keyword search techniques in smart city. EURASIP J Wirel Commun Netw. 2021; 2021. https://doi.org/10.1186/s13638-020-01875-2 .
3. Yuan X, Wang K, Lin J, Wang C, Yu PS. Privacy-preserving Deep Learning with SPDZ. Neur IPS. 2020. https://doi.org/10.48550/arXiv.2010.09582
4. Chen J, Dai W, Shi R, Li T, Weng J. HETE: Heterogeneous Autoencoder for Encrypted Textual Search in Cloud. IEEE Trans Parallel Distrib Syst. 2022; 33(3): 588-600. https://doi.org/10.1109/TPDS.2021.3111425
5. Li T, Chen J, Li Z, Weng J, Lee K, Deng RH. Building Confidential and Efficient Query Services in the Cloud with RASP Data. IEEE Trans Knowl Data Eng. 2020; 32(1): 141-153. https://doi.org/10.1109/TKDE.2018.2881661
6. Zhang Q, Luo T, Reiter MK. HEDGES: efficient homomorphic encryption based PIR for distributed log data. In: Proc ACM SIGSAC Conf Comput Commun Secur. 2019 (pp. 1583-1600). https://doi.org/10.1145/3319535.3363241
7. Brasser F, Sadeghi AR, Schneider T, Wehrenberg I. CHAMELEON: A Hybrid Secure Search Scheme over Encrypted Data. ACSAC 2019. https://doi.org/10.1145/3359789.3359790
8. Fu M, Ren K, Shao J, Zhang C. Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement. IEEE Trans Inf Forensics Secur. 2019; 15: 1345-1358. https://doi.org/10.1109/TIFS.2019.2948660
9. Xu P, Jin H, Wu Q, Wang W. Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack. IEEE Trans Comput. 2019 Nov 11; 68(11): 1630-42. https://doi.org/10.1109/TC.2019.2916786
10. Zhang Y, Chai Z, Fan P, Li K. Enabling efficient multi-keyword search with strong privacy over encrypted data in cloud. Concurr Comput Pract Exp. 2018 Feb 15; 30(17): e4321. https://doi.org/10.1002/cpe.4321
11. Chen Z, Luo C, Li S, Zhang Q, Chen L, Luo X. Quasi-Fully Homomorphic Encryption Scheme for Privacy-Preserving Neural Network. IEEE Trans Comput. 2022 Feb 14. https://doi.org/10.1109/TC.2022.3144433
12. Rathee D, Iyer RR, Chandran S, Gonggrijp R, Gupta I. Cryptflow2: Practical 2-party secure inference. In: Proc ACM SIGSAC Conf Comput Commun Secur. 2020 Oct (pp. 525-538). https://doi.org/10.1145/3372297.3417886
13. Yi X, Paulet R, Bertino E. Homomorphic encryption experiments on IBM's cloud quantum computing platform. IEEE Trans Emerg Top Comput. 2020 Apr 16.: https://doi.org/10.1109/TETC.2020.2986492
14. Chu C, Deng J, Zamani H, Wu J, Li S. Image retrieval with split augmented convolutional features. In: Proc

Published Online First: June, 2024
https://doi.org/10.21123/bsj.2024.10215
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

IEEE/CVF Int Conf Comput Vis. 2019 (pp. 9543-51).project n.d. (accessed February 3, 2024).
15. Zhan H, Sheng VS. Privacy-Preserving Representation Learning for Text-Attributed Networks with Simplicial Complexes. Proc AAAI Conf Artif Intell. 2023; 37: 16143–4. https://doi.org/10.1609/aaai.v37i13.26932
16. Li T, Xiao X, Lu Y, Zhou L. Search me if you can: privacy-preserving location query over encrypted data. In: Proceedings of IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. https://doi.org/10.1109/INFOCOM.2018.8485905
17. Wotaifi TA, Dhannoon BN. An Effective Hybrid Deep Neural Network for Arabic Fake News Detection. Baghdad Sci. J. 2023; 20(4): 1392-1401. https://doi.org/10.21123/bsj.2023.7427
18. Esquivel JA, Li D. Lightweight and Personalized E-commerce Recommendation based on Collaborative Filtering and LSH. International Journal of Ad Hoc and Ubiquitous Computing 2024;45. https://doi.org/10.1504/ijahuc.2024.10059174 .
19. Hassan FO, Samir NM, Hanapi ZM. Impacts of Denial-of-Service Attack on Energy Efficiency Pulse Coupled Oscillator. Baghdad Science Journal 2023. https://doi.org/10.21123/bsj.2023.7161 .
20. Yang R, Wang S, Gu Y, Wang J, Sun Y, Zhang H. Continual learning for cross-modal image-text retrieval based on domain-selective attention. Pattern Recognition 2024; 149:110273. https://doi.org/10.1016/j.patcog.2024.110273 .
21. Yadav C, Yadav V, Kumar J. Secure and Reliable Data sharing scheme using Attribute-based Encryption with weighted attribute-based Encryption in Cloud Environment. International Journal of Electrical and Electronics Research 2021;9:48–56. https://doi.org/10.37391/ijeer.090305 .
22. A. A, Saleh S, Elkafrawy P. Searching Techniques over Encrypted Cloud Data. International Journal of Computer Applications 2019;181:25–30. https://doi.org/10.5120/ijca2019918676 .
23. Cui S, Song X, Asghar MR, Galbraith SD, Russello G. Privacy-preserving Dynamic Symmetric Searchable Encryption with Controllable Leakage. ACM Transactions on Privacy and Security 2021; 24:1–35. https://doi.org/10.1145/3446920 .
24. Zhu D, Zhu H, Wang X, Lu R, Feng D. Efficient and Privacy-Preserving Similar Patients Query Scheme Over Outsourced Genomic Data. IEEE Trans Cloud Comput. 2023; 11: 1286–302. https://doi.org/10.1109/tcc.2021.3131287 .
25. Lu H, Chen J, Ning J, Zhang K. Verifiable Conjunctive Dynamic Searchable Symmetric Encryption With Forward and Backward Privacy. Comput J. 2022; 66(10): 2379–92. https://doi.org/10.1093/comjnl/bxac084 .

تحسين البحث المشفر في السحابة باستخدام تقريب الاستعلام القائم على الترميز الذاتي

محمود محمد، خالد الاعثمان

جامعه الملك عبد العزيز، كليه الهندسة والحاسبات، المملكة العربية السعودية.

الخلاصة

يتيح البحث المشفر الاستعلام الخاص عبر البيانات السرية. ومع ذلك، فإن التشفير يأتي بتكلفة باهظة للكفاءة. يوفر هذا البحث تحليلاً موسعاً لاستخدام تقريب الاستعلام القائم على الترميز الذاتي لتحسين سرعة ودقة البحث المشفر في السحابة. من خلال التجارب الشاملة على مجموعات البيانات الحقيقية، نقدم رؤى جديدة في تحسين النموذج واستراتيجيات التدريب والمبادلات الناشئة عن النشر. توجه نتائجنا تحسين الصلة والكفاءة والأمان .نقترح إطار عمل يقوم بتدريب مرمز ذاتي على مجموعة البيانات لترميز السجلات في تمثيل مكثف. وقت البحث، يتم تحويل استعلامات المستخدمين إلى نفس الفضاء الكامن حيث يتم حساب التشابه تحت التشفير، مما يتجنب فك التشفير المكلف .تحلل منهجيتنا بدقة كيف تؤثر العوامل المعمارية وتقنيات التدريب ومجموعات البيانات ومخططات التشفير على الأداء من الطرف إلى الطرف الآخر والقابلية للتوسع على البنية التحتية للسحابة. نحدد أفضل الممارسات التي تعظم الدقة والسرعة مع الحفاظ على الخصوصية. يتم إجراء تحليل مقارن شامل مع طرق بديلة مثل TF-IDF و LSI و BM25 على الفهارس المشفرة. يؤسس عملنا تقريب الاستعلام القائم على الترميز الذاتي كحل عملي جاهز للنشر على نطاق واسع في السحابة.

الكلمات المفتاحية: المرمزات الذاتية، الحوسبة السحابية، البحث المشفر، تقريب الاستعلام، استرجاع المعلومات.