Integrating BERT-Based Feature Extraction with Traditional Algorithms for Low-Latency DNS Mappings in Osmotic Computing

MSc Research Project
Cloud Computing

Meerath Nida Aman
Student ID: x21228841

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Meerath Nida Aman
Student ID:	x21228841
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	14/12/2023
Project Title:	Integrating BERT-Based Feature Extraction with Traditional Algorithms for Low-Latency DNS Mappings in Osmotic Computing
Word Count:	5755
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Meerath Nida Aman
Date:	28th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Integrating BERT-Based Feature Extraction with Traditional Algorithms for Low-Latency DNS Mappings in Osmotic Computing

Meerath Nida Aman
x21228841

Abstract

This work presents a novel technique to enhance the recognition and classification of MicroElement (MEL) in Osmotic Technology. The solution integrates traditional clustering algorithms with state-of-the-art Natural Language Processing (NLP) techniques to optimize resource utilization and minimize latencies. The system utilizes a transformer-based model known as Bidirectional Encoder Representations from Transformers (BERT) to extract features from DHCP databases. It then enriches structural components such as hops, latency, and geolocation with contextual data that is more nuanced. The proposed composite grouping methodology seamlessly integrates traditional techniques such as agglomerative clustering and K-Nearest Neighbours (KNN) with BERT-based semantic modeling. Our objective is to provide a comprehensive understanding of Fully Qualified Domain Names (FQDNs), resulting in the implementation of intelligent clustering that is both semantically and architecturally advanced. The method's focus on achieving low-latency Domain Name System (DNS) translations is a noteworthy advance. By taking into account the inclusion of hops latency and geographical location, this technique aims to enhance the efficiency of DNS translation in various computational configurations. The integration of semantic complexity from BERT enhances the adaptability to evolving computing environments, hence enhancing the overall efficiency and efficacy of Osmotic Processing. This paper presents a complete approach that combines state-of-the-art NLP techniques with traditional clustering algorithms to address the challenges of Osmotic Computing environments. This signifies a significant advancement in the realm of decentralized computer paradigm.

1 Introduction

Osmotic computing is a framework for distributed computing that uses Micro Elements (MELs) to make it easy to move and deploy apps across cloud, edge, and IoT nodes. MELs are small, self-contained services that can be installed and controlled in a transparent way in multiple software layers. This makes it possible to launch and scale applications in a way that makes the best use of both performance and resources. The effective translation and management of FQDNs is important for systems to work more efficiently in the field of Osmotic Computer Science, where discovering and gathering MELs has become more important for making the DNS faster and more efficient at sharing resources. In this research a new method that combines cutting-edge NLP techniques, like BERT, with more traditional clustering methods, like KNN, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), and agglomerative clustering is implemented.

This research is based on the need for a new approach that can solve these challenges that come up in Osmotic Computing settings. The suggested method is unique because it combines BERT-based feature extraction with well-known clustering algorithms such as KNN and agglomerative grouping in a way that does not affect the results. This mixed method uses both architectural classification and semantic feature extraction to get a full picture of DNS information. The clustering methods create groups with more detailed features from BERT by taking into account contextual differences between FQDNs and structural similarities.

The primary objective of this unique combination is to make low-latency DNS translations easier, which will lower latency and make Osmotic Computing systems more efficient overall. Optimizing resource utilization is very important and by expertly routing requests to appropriate MELs, we can cut down on latency and improve total resource efficiency. This would mean that apps will run faster and respond more quickly, especially those that can use MELs close to users or IoT devices. The decentralized nature of osmotic computing makes it clear that MELs are not controlled centrally. This highlights how important it is for users to be able to easily find and use the services they need for their computing needs. The hops, delay, and geolocation are considered during the clustering process to make sure that the translations that come out are context-aware and useful, which fits with Osmotic Computing's focus on flexibility. The fact that hops, latency, and geolocation are all taken into account during the clustering process shows that the algorithm is dedicated to providing minimal latency DNS translations and changing MEL findings in the context of decentralized computing.

Through a number of different approaches, this paper makes significant additions to the body of scientific literature. First, it presents a new way to find MEL that uses NLP and clustering techniques in a way that has never been done before. This new method looks like it could provide a complete and complex answer to the problems that come up with identifying MEL in osmotic computing. This study also talks about the creation of a complex algorithm that uses BERT to get features from DNS records. This algorithm is meant to improve the accuracy and amount of information gathered from DNS data, which will help us understand more about MEL characteristics. Lastly, the study adds to the scientific discussion by using experimental data to carefully evaluate the suggested method. Looking at how traditional clustering methods and NLP can be used together in this study is an optimistic approach for pushing osmotic technology forward.

Research Niche: This study is the first to use BERT, a transformer-based method, to get nuanced data from DNS databases. It introduces an innovative way of approaching Osmotic Computation. This method makes it easier to understand FQDNs in the Osmotic Computation ecosystem by adding contextual information to common measures like hops and latency. The study adds to existing clustering methods with the combination of agglomerated clustering and KNN, which would make BERT-based feature discovery work better. This new combination makes FQDN clustering smarter and more aware of context by using semantically encoded data from BERT to improve the grouping process and get better structural insights.

Research Question: How can the combination of BERT-based semantic modeling and traditional clustering algorithms be effectively harnessed to optimize the identification and clustering of MELs in the context of Osmotic Computing within larger geographical regions?

2 Related Work

In Osmotic Computing, where the combination of different technologies is changing the way distributed computing is done, it is necessary to carefully examine the existing literature. This literature review not only shows how important MEL discovery and organizing are in Osmotic Computing, but it also takes a close look at the methods and novel proposals proposed by prominent researchers . The primary goal of this literature review is to look into artificial intelligence (AI), NLP, DNS, and new methods like the Osmotic Computational Enhanced Domain Name Systems (OCE-DNS) method.

2.1 Tensor-Based Neural Networks in AI for MEL Recognition

Researchers have found that using Tensor-based neural networks as a base for AI in the discovery of MELs is a big step forward in understanding how AI works at its most basic level. The paper Abadi et al. (2016) set the stage for neural networks to be used in many different ways, which makes them a strong tool for solving problems in MEL recognition. The weakness, though, is that there aren't any specific examples of how Tensor-based neural networks help with MEL recognition. It would be helpful for the research if there were real-life examples of how these networks can be used in Osmotic Computing.

2.2 NLP Approaches for Contextual FQDN Comprehension

Machine learning, or NLP, is the key link between information that computers can understand and knowledge that people can read. Decentralized representations of concepts and words Mikolov et al. (2013) and featured clustering techniques for categorizing texts Dhillon et al. (2003) make it possible for machines to understand and handle text data well. By using NLP techniques, it becomes easier to understand how domain names fit into the bigger picture of collecting FQDNs. This makes MEL classification and forecast more accurate. Because it is close to both end users and tools, fog computing is a key way to make the best use of computer resources in many situations. The study by Yi et al. (2015) gives useful details about the ideas, uses, and problems that come up when designing fog-based computing. This is in line with the flexible and situation-sensitive approach that was suggested for MEL development. This match shows that MEL the finding is adaptable and works well in many situations, which is important for improving Osmosis Computing's efficiency.

2.3 Big Data Analytics and Machine Learning in Osmotic Computation

The creation of the Osmotic Computational Enhanced Domain Name Systems (OCE-DNS) method is a key part of this study. It is a big step forward in domain registration systems. The Osmotic Computing Consortium (2021) says that the OCE-DNS technique is a more flexible and situation-aware way to resolve domain names. This plan fits with the need for smart FQDN grouping. It offers a way to change traditional DNS servers and lower access time and hop counts, which makes Osmotic Computing work better. Managing huge amounts of data is always a part of distributed processing. This is where big data analytics and machine learning meet. The study shows how important it is to have flexible and effective ways to deal with the large amounts of data that come with Osmotic Computation. It also looks at how important it is to use big data analysis along with machine learning techniques Gandomi and Haider (2015). The research by Puthal et al. (2015) into the features and issues of computing in the cloud highlights the need for reliable solutions for handling huge amounts of data by revealing potential issues and growth areas.

2.4 IoT and Fog Computing for Osmotic Efficiency

Because more and more IoT devices are being used in spread-out computer environments, networking options that work with osmotic computing are needed. A study by Puliafito et al. (2019) says that fog technology for the IoT helps us understand the issues and possible solutions in this area. The main goal is to make Osmotic Computing more useful in a range of changing situations and to make the best use of resources. This is similar to how fuzzy computing and neural networks work together.

2.5 BERT-Based Feature Extraction in DNS Mapping

Since machine learning, NLP, fog computing, and advanced domain name systems are all coming together, the way MELs are found and used is changing. These methods not only solve the problems that come with computer science that is spread out, but they also open up a completely new, more effective, and adaptable way for the Osmotic Computation environment to move forward. Combining improved clustering methods with standard algorithms and BERT-based feature extraction for low-latency DNS maps is a cutting edge area of research. Because it is flexible and works well, BERT-based extracted features can be added to traditional clustering methods to make DNS maps more accurateSu and Su (2023). Barabási and Albert (1999) explanation of network science ideas helps us understand new patterns of growth in random networks. By using these new ideas to make clustering methods better, DNS translations in the setting of Osmotic Computers might become more stable and useful. These methods could be made bigger and better by adding BERT-based featuresLe et al. (2022). This could help show complicated patterns in DNS data better and make it easier to tell them apart. The information-theoretic features clustering method for text categorization by Dhillon et al. (2003) gives a new way of looking at clustering approaches. Enhanced clustering might be able to tell the difference between different DNS maps if these methods are combined with BERT-based extracted features. This would make it better at telling them apart.

2.6 Osmotic Computing Enabled-Domain Name System (OCE-DNS)

Osmotic Computing Enabled-Domain Name System (OCE-DNS), a geocode-based name system for osmotic computing, is proposed in this study for better discovery of MELsGalletta et al. (2021). OCE-DNS stores geocodes in a distributed RR database and changes the DNS records on the fly to reflect MEL migration. This lets people across the SDMem easily and quickly get to MELs and apps.

3 Methodology

To improve MEL recognition in Osmotic Computing, a new method is being suggested that uses a combination of advanced NLP techniques and well-known clustering algorithms. The first important step in the execution process is to identify features and prepare the data. It is used here to improve the meaning of FQDNs data using BERT. BERT's contextual data collection tools give us a more complete picture of the relationships inside FQDNs, which makes it possible to do more in-depth research later on.

After the feature extraction, BERT creates embeddings that act as enhanced characteristics, which give further clustering process more depthAlaparthi and Mishra (2020). It carefully combines traditional clustering methods like agglomerative clustering and KNN, using the improved embeddings to create more intelligent groups of MELs based on their context. It's carefully explained why these standard algorithms should be used based on the job at hand and the results that are wanted for MEL discovery in Osmotic Computing settings.

In the Design and Specification step, metrics that are important to the MEL recognition goals are set and explained. It goes into great detail about how the experiment was set up, including the hardware requirements, software dependencies, and settings for both BERT and the clustering algorithms. In the Implementation and Evaluation stages, the method is put into action step by step, from feature extraction to clustering, and then the results are carefully analysed in the results and discussion stage.

The results of this study show how the suggested way makes it much easier to find MELs in the Osmotic Computing framework. The conclusion talks about the main points of the study and focuses on what the improved MEL recognition means for the Osmotic Computing model and what the future scope.

4 Design Specification

In the Figure 1 the high-level architecture diagram represents a complete sequence of data transformations that are essential for the effective functioning of the system. The process begins with data loading, wherein the unprocessed data is ingested into the system. After that, a sequence of intricate steps in data preprocessing are carried out, which include tasks such as HTML parsing, label encoding, splitting FQDN data, tokenizing, stemming, and padding. Each of these steps together contribute to refining and structuring the data in a way that is suitable for analytical processes that follows. A crucial stage in the architecture involves using the BERT transformer to extract data. BERT, well-known for its ability in dealing with NLP problems, is used to understand and capture the nuanced contextual details within the data. In this stage, BERT generates attention masks, which increase the understanding of the interdependencies within the data. After finishing the process of data preprocessing and extraction, the architecture focuses on using clustering

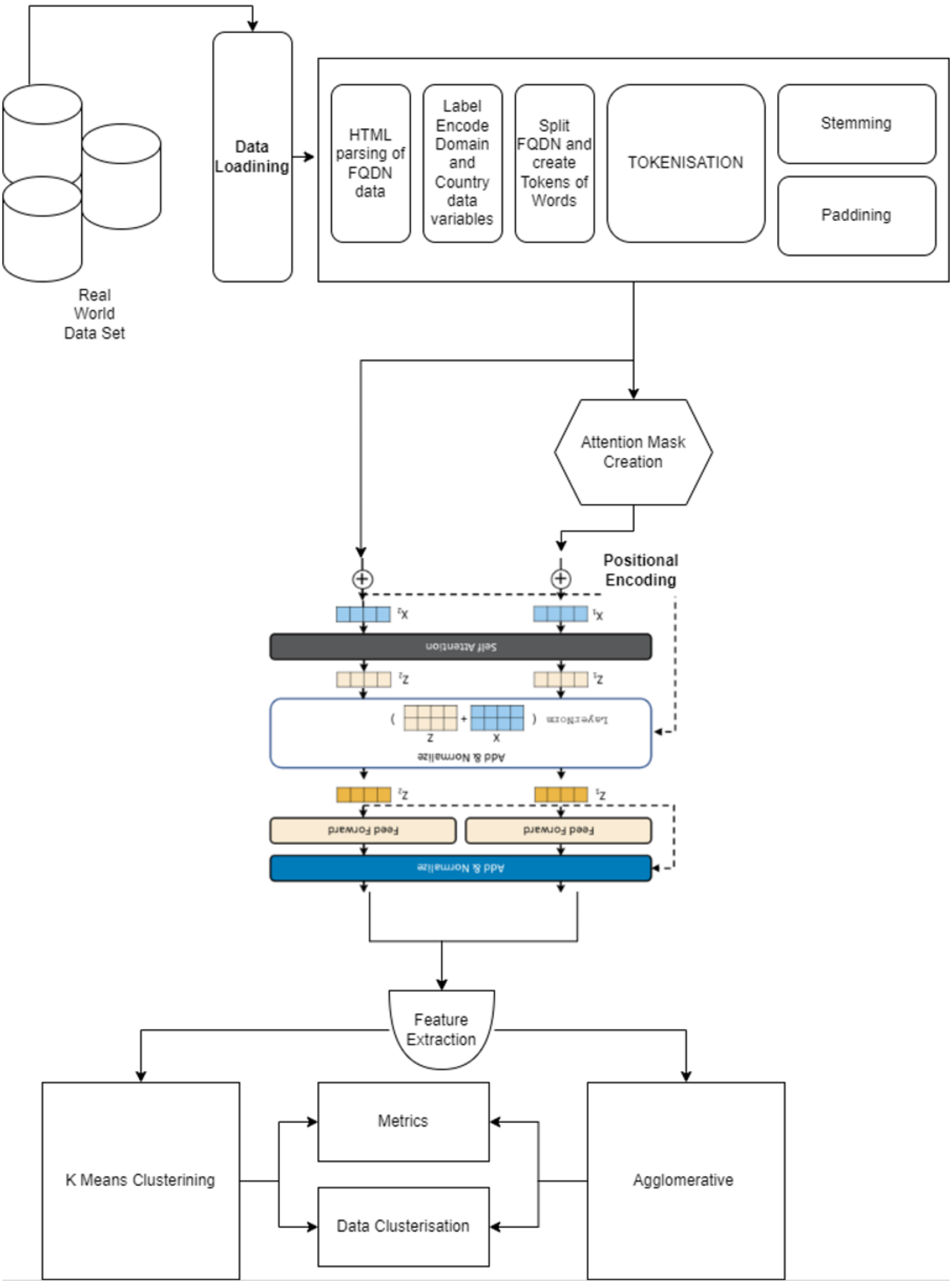

Figure 1: High Level Diagram of Clustering of DNS

algorithms. These techniques are important in organizing and categorizing data based on inherent patterns and similarities.

4.1 Bidirectional Encoder Representations Using Transformers

BERT, which stands for "bidirectional encoder representations using Transformers," is a powerful NLP model that can find contextual links in textual data. BERT is used to get meaningful information from DNS FQDNs, as shown in Figure 1. While most approaches only look at structural aspects, BERT takes into account the complex relationships that are stored in the FQDNs. BERT preserves the broader significance of FQDNs even while using industry-specific jargon or complicated subdivisions. This lets us understand the semantic makeup of FQDNs better. This better understanding makes it easier to extract features that can be used in later clustering techniques. This gives the DNS data a more comprehensive depiction. The ground breaking NLP paradigm BERT can extract contextual connections from textual data, which is one of its most interesting featuresAlaparthi and Mishra (2020).

4.2 Clustering techniques

KNN is a flexible clustering technique that is often used to solve architectural issues in classification. KNN is used with DNS FQDNs to cluster related FQDNs together based on basic factors such as hops, latency, and locationXu et al. (2011). KNN finds and clusters certain FQDNs based on how similar they are between network hops or proximity in space. This technique does a great job of finding the essential similarities between FQDNs. It then creates coherent groups that show how DNS components are structured and distributed across the world.

The DBSCAN method is a well-known clustering method that can identify clusters of different sizes and shapes. DBSCAN is essential for identifying anomalies and confusion in the DNS data, as these could lead to unusual or possibly dangerous FQDNsCui et al. (2014). If some FQDNs show strange patterns in regarding connections or latency, DBSCAN could identify these strange patterns as cacophony spots. This can help enhance security protocols in Osmotic Computing settings.

Agglomerated clustering is a hierarchy technique that builds an ordered network of linkages by repeatedly merging comparable points of data. When setting up FQDNs of DNS, clustering by agglomeration is used to show relationships deep inside the data set. Agglomerative aggregation captures hierarchy linkages and subclusters that are present in certain FQDN groupsMonath et al. (2021). Through this technique, we can better understand how FQDNs are organized by highlighting the links and connections that appear throughout the DNS information.

Using BERT, KNN, DBSCAN, and agglomerative clustering techniques makes it easier to look at FQDNs of DNS in greater detail in the context of Osmotic Computing. To quickly find and cluster MELs, BERT enhances extracted features with knowledge of context, and algorithms for clustering give hierarchical and topological insights. Together, these methods enable a complete methodology.

4.3 Dataset Description

The dataset comprises essential attributes for DNS analysis:

- FQDN: Fully Qualified Domain Name.
- IP: IP address associated with the FQDN.
- Location: Geographic location information.
- Hops: Number of network hops required to reach the destination.
- Latency: Time delay between sending and receiving a response.
- Response Time: Overall response time for the DNS query.

The sample dataset is shown in the Table 1.

FQDN	IP	Location	Hops	Latency	ResponseTime
gazetteandherald.co.uk	36.108.22.65	Los Angeles	8	29.51	49.78
mlssoccer.com	151.203.156.167	Sydney	1	78.86	15.55
reddit.com	4.21.169.144	Los Angeles	9	88.19	30.66
redbullracing.com	56.240.90.237	Sydney	5	33.66	30.57
stackexchange.com	200.76.224.26	Dublin	4	48.58	21.03
fifa.com	37.133.159.54	Hyderabad	5	70.28	35.08
sufc.co.uk	139.115.241.174	Berlin	2	29.38	16.79
economist.com	136.121.238.48	Los Angeles	6	83.22	46.21
vulture.com	7.0.146.14	Dublin	1	22.13	39.1
refinery29.com	104.117.134.233	Dublin	10	93.68	15.69

Figure 2: Table 1 containing Dataset Attributes

5 Implementation

This implementation uses cutting-edge NLP and machine learning techniques to analyze DNS data. Utilizing libraries like Transformers and scikit-learn, the script processes FQDN through advanced methods, including BERT for feature extraction and diverse clustering approaches. The result is a sophisticated analysis pipeline intended to uncover meaningful patterns within the DNS dataset.

5.1 Library Imports

The script initiates by importing important libraries including Transformers, NLTK, TensorFlow, scikit-learn, and NumPy. The fact that these libraries are essential for NLP and machine learning tasks suggests that the script probably makes use of sophisticated modelling and processing methods. A lot of NLP tools are in NLTK, and TensorFlow is a powerful framework for machine learning jobs. Scikit-learn is a versatile library with many machine learning algorithms, and NumPy provides support for numerical operations. Putting these libraries together suggests a sophisticated method that uses advanced tools for complex data handling.

5.2 Loading DNS Data

The script loads DNS data from a CSV file to a Pandas DataFrame, which is a well-known Python framework for data manipulation. This dataset is expected to contain information about FQDN and their associated located. The 'FQDN' field serves as the main identifier for future transformations.

5.3 Label Encoding and Column Creation

Once the data is loaded, label encoding is used to turn the descriptive geographical names into numerical representations. This is very important for machine learning systems that need numerical input. Based on the "FQDN" field, two more fields are made: "FQDN Sentence" and "Domain." 'Domain' gets information about domains, and 'FQDN Sentence' gets information about sentences that come from FQDNs. The encoding of these groups prepares the data for later examination.

5.4 Transfer Learning with BERT

BERT is a transformer-based model that has already been trained and is known for how well it understands contextual information in natural language. The code uses transfer learning with BERT. Before the FQDN words can be put into the BERT model, they need to be tokenized and an attention mask needs to be generated. The code can correctly gather the subtle connections and meanings in the FQDN data by using BERT.

To use BERT as a feature extraction tool for DNS segmentation in Osmotic Computers, a number of hyperparameters need to be set that determine how it is structured and trained. The base is the model design, which has details about the type of BERT variant (Base, Large, etc.), the number of hidden layers, the focus heads, and the hidden size. The number of training epochs, batch size, learning rate, and maximum sequence length are some of the other things that have a significant impact on the model efficiency. those hyperparameters are standard, the values of the parameters may be different based on the version of BERT that is being used and how complex the DNS identification job is. When these hyperparameters are changed to fit the specifics of the dataset, models will be more accurate and the extraction of attributes from web addresses are done more quickly.

The arrangement of variables of a BERT model are set by the BertConfig class in Table 2. These variables are contained in the table. In both training and speculation, each parameter is very important for determining how the model is built and how it behaves. BertForMaskedLM was chosen because it works well with masked language modeling problems. In these problems, the model is taught to guess words that are missing from phrases. Notably, the failure probability for both the concentration probability and the hidden layers is set on purpose at 0.1. This is done to avoid overfitting during training. It looks like a focused regularization approach based on the null result, which means there were no dropouts in the classifier layer. There is also a hidden size of 768 in the configuration. This sets the number of dimensions for internal representations and requires the GELU activation method for hidden layers. Other important factors that affect the model's depth, multi-head attention to oneself, and token processor skills are the number of focused heads (12), the number of hidden layers (12), and the size of the vocabulary (30522). Some of the things that affect the structure and training of

Configuration Parameter	Value
Architectures	BertForMaskedLM
Attention Probs Dropout Prob	0.1
Classifier Dropout	null
Gradient Checkpointing	false
Hidden Activation Function	gelu
Hidden Dropout Prob	0.1
Hidden Size	768
Initializer Range	0.02
Intermediate Size	3072
Layer Norm Eps	1e-12
Max Position Embeddings	512
Model Type	bert
Number of Attention Heads	12
Number of Hidden Layers	12
Pad Token ID	0
Position Embedding Type	absolute
Transformers Version	4.30.2
Type Vocab Size	2
Use Cache	true
Vocabulary Size	30522

Figure 3: Table 2 containing BERT Hyperparameters

the model are the starting range for the weights, the use of gradient checkpointing, and a number of factors related to layer normalization and position embedded data. Using the transformers library version (4.30.2) makes sure that everything is consistent and fits with the specified architecture. This complete setup gives a way to build a BERT model for a certain NLP job that takes into account the model's regularization, design details, and hyperparameters.

5.5 Clustering Methods

After extracting features with BERT, the script uses clustering methods to group FQDNs that are similar together. Some of the methods that are used are KMeans, DBSCAN, and clustering by agglomeration. The elbow method is used to find the best amount of clusters for KMeans. This gives the clustering process a quantitative basis. The implementation includes using the KNN method to figure out how distinct or similar points of data are based on their BERT-derived attributes. For each of its k close neighbors, the KNN gives the information point to every other class. The parameter k controls how many neighbors are taken into account during the decision-making process. By comparing how similar two or more close data points are, this method can be used to find clustering in the BERT feature space.

In contrast, Agglomerative Clustering is a hierarchical clustering method that repeatedly merges clusters based on a linkage criterion after initialising each data point as a separate cluster. During the merging phase, the script measures the similarity across clusters using the high-dimensional features produced by BERT in the overall setting of BERT feature extraction. The hierarchy in agglomerative clustering depicts clusters in a more organized way, showing connections between data points and gives in depth understanding of the underlying trends.

In addition, a technique known as DBSCAN is implemented. DBSCAN is very good at recognizing different shapes and density of clusters. Based on the attributes that BERT gives, DBSCAN probably finds areas with a high density of data points to identify clusters. This method makes clustering more flexible because it does not need the total amount of categories to be specified in advance. It is also good at finding outliers.

5.6 Evaluation Metrics

The script uses metrics like silhouette scores to judge the strength of clustering. Silhouette scores tell us how similar an object is to its own cluster compared to other clusters. This gives us a way to measure how well clusters hold together. These measures help make sure that the clustering methods used are good at organizing the data.

5.7 Visualization

Scatter plots are used to show the results of the grouping. Scatter plots make it easy to see how the relationships and groups found in the dataset are connected. This visual review helps to make sense of the clustering results, which leads to a complete understanding of the patterns present in the DNS data.

5.8 Recurrent Investigation

The script shows that variables and techniques for grouping are being investigated again and again. To find the most important clusters in the dataset, this iterative technique suggests trying out a lot of different parameters and methods. It shows a commitment to an in-depth analysis that will reveal the deepest patterns and trends in the DNS data.

6 Evaluation

In this Section the results of clustering that was achieved by combining BERT-based semantic modeling with standard clustering methods are evaluated and discussed. Evaluation metrics, like silhouette scores and the elbow method, are used to figure out how effective the clustering results are and how many clusters should be used. This section's objective is to give a thorough look at how well the proposed method improves the finding of MELs, especially within a larger geographical context, and how it helps in the advancement of Osmotic Computing.

Silhouette analysis is a good way to evaluate the quality of the clusters that a clustering method creates. In Osmotic Computing, domain-related characteristics are first put through BERT embeddings, which make it possible for further clustering when it comes to DNS categorization. Following KMeans grouping with the optimal K value found, the silhouette's score is produced. Higher scores mean that the groups are deeper and more distinct. This score indicates degree to which groups are. So, the silhouette distance calculation gives a number that shows how well the clustering works, which helps with the evaluation of the DNS classifying algorithm.

The StandardScaler from scikit-learn is used to make sure that each feature has an average of 0 and an average variation of 1. Standardization makes all of the traits the same size so that one does not stand out too much from the others because of differences in magnitude which is very important while using clustering techniques. The standardized characteristics are shown on a three-dimensional scatter chart with lines on a grid in Figure 4. A single point of information from the simulated data set is represented by each point in the figure. The data is shown spatially because the features are plotted on the X, Y, and Z planes.

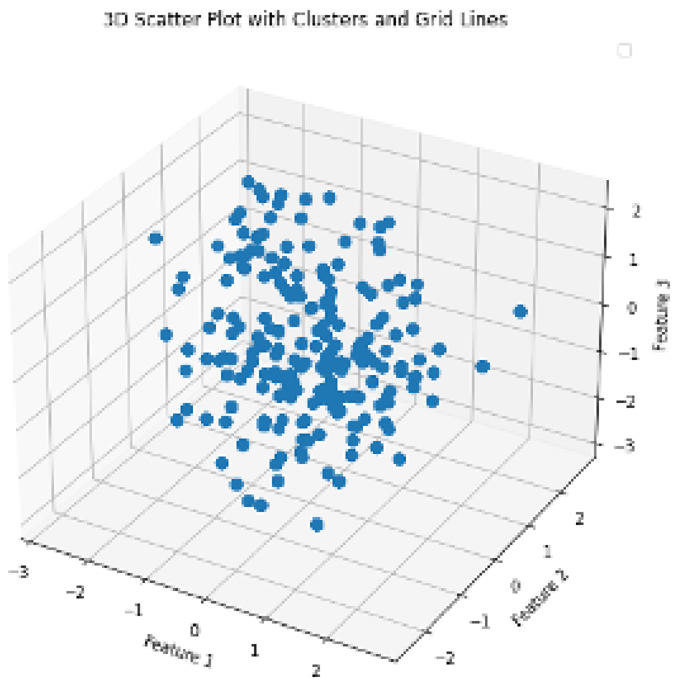

Figure 4: Feature Parametric Visualization across 3 Dimensions

6.1 Experiment 1

The elbow method is a way to determine how many clusters should be in a dataset. Different values of K , which are the number of clusters, are used to run a clustering algorithm like KMeans and a measure of clustering quality is calculated for each K. After the features are extracted using BERT, the KMeans method is used, focusing on K=5 and K=7 clusters (Figure 6 and Figure 7). The elbow method is used to plot the total squared lengths (inertia) against the total number of groups for different K values as shown in Figure 5. In this graph, the "elbow" point where model complexity and clustering efficiency meet is where the ideal K value is. Implementing this approach allows finding the right number of clusters for further investigation.

The silhouette scores for K=7 are 0.1659 and for K=5, they are 0.1555. A higher silhouette score means that the clusters are more distinct. The values show how closely the clusters are interconnected and how distinct they are. The K values of the hyperparameters tell how many clusters to divide the information into. In terms of DNS clustering, this could be seen as putting FQDNs into different groups based on how similar their features are, which could mean that they are functionally or geographically close.

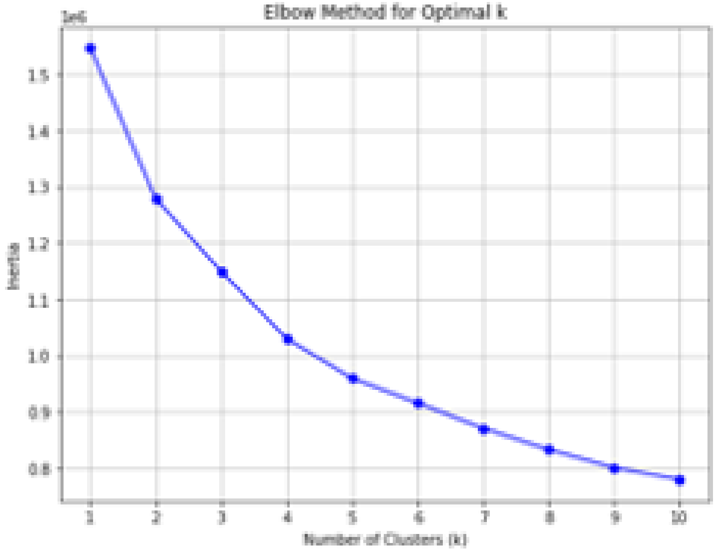

Figure 5: KMEANS elbow methods for optimal clusters

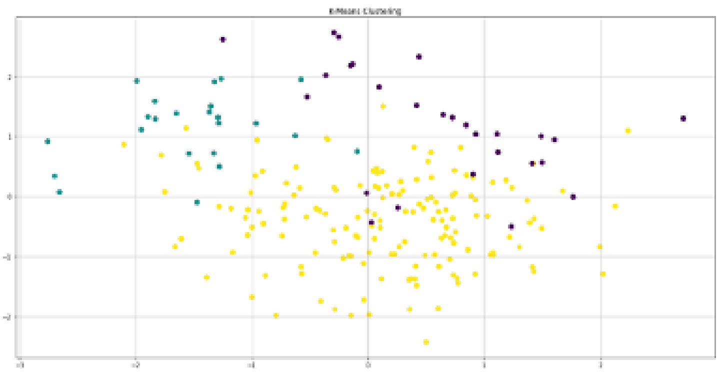

Figure 6: KMeans Clusters at K = 5

6.2 Experiment 2

DBSCAN does not need a predetermined number of clusters because it uses density-connected areas instead. The total silhouette score for DBSCAN shown in Figure 8 is 0.1393, which is about the same as the number for the Agglomerative Clustering of K=5 silhouette score. Clusters are made by DBSCAN based on densely connected areas. Parameters like min-samples, which controls the smallest number of points in a neighborhood, and epsilon, which controls the neighborhood radius, are very important. This method can be used to find FQDNs that are closely connected to DNS groups, which could point to places with efficient communication channels and low latency.

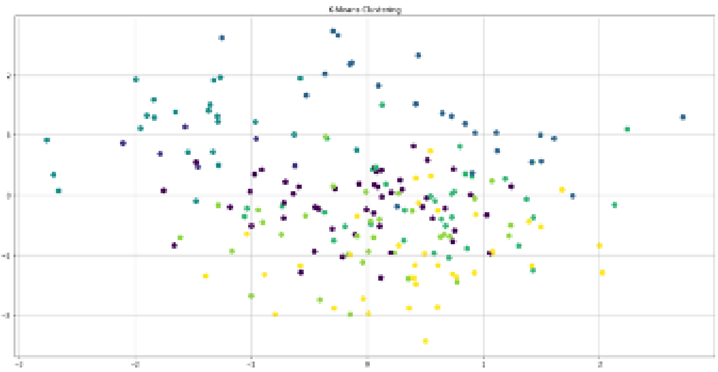

Figure 7: KMeans Clusters at K = 7

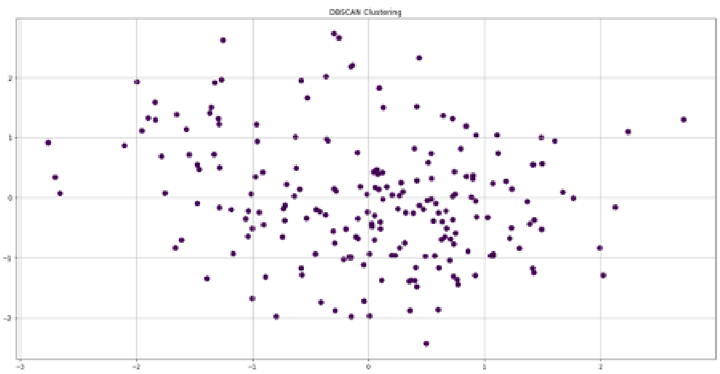

Figure 8: DBSCAN Cluster

6.3 Experiment 3

Using a hierarchical method, agglomerative clustering groups pieces of information together based on linking techniques. The silhouette scores for the different levels of K - 3, 5, and 2 are 0.2121, 0.1415, and 0.2930, respectively shown in Figure 9, Figure 10 and Figure 11. Lane is the linking technique that reduces variation within clusters. Notably, K=2 has the highest silhouette score, which means that the clusters are clear and distinct. When it comes to DNS aggregation, these clusters could be made up of FQDNs that are connected in similar ways. This could help lower latency and hop counts so that clients can communicate more effectively.

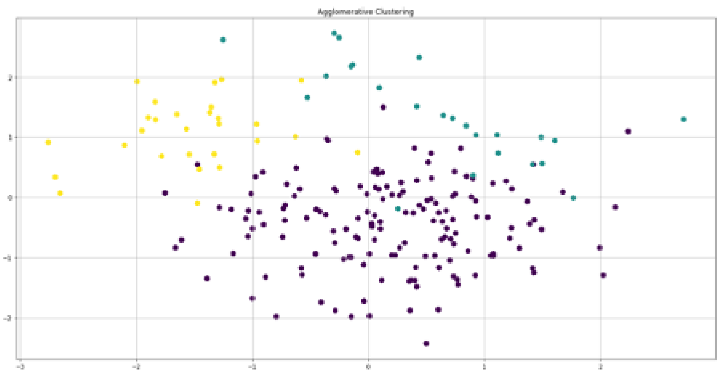

Figure 9: Agglomerative Clustering at N = 3

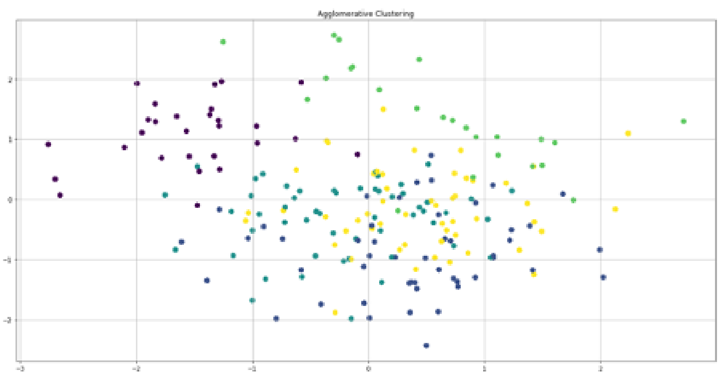

Figure 10: Agglomerative Clustering at N = 5

Comparing the results in experiments 1,2 and 3 for clustering, we see the DBScan performance has shown severe deterioration while predicting only a single cluster. However the KMeans and Agglomerative clustering techniques show promising results for the clustering problem. The Table 4 discusses in detail the hyperparameter's and the algorithms in detail.

Table 4 containing the results of all three clustering algorithms, KMeans, Agglomerative, and DBSCAN shows how well each algorithm can reduce communication latency and increase hop count to make connections between clients faster. Each one has its own way of splitting the FQDN information into clusters. To understand how these al-

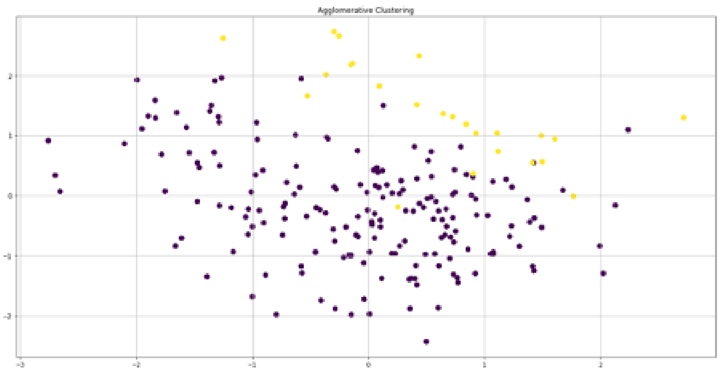

Figure 11: Agglomerative Clustering at N = 2

gorithms could be used in the setting of worldwide DNS groups for better interaction, it is important to take a critical look at them and change their hyper parameter settings.

7 Conclusion and Future Work

In conclusion, this research shows how important it is to lower latency and make the number of hops as small as possible to fasten the communication in Osmotic Computing. A key part of reaching this goal is finding a link between clustering results and BERT for transformer-based DNS or FQDNs. As a powerful transformer-based model, BERT is very helpful for getting out FQDNs and other contextualized text-based data. The quality of the attributes extracted by BERT determines how well clustering methods work at grouping FQDNs. This, in turn, is a key factor in reducing latency and finding the best communication paths.

To answer the research question, the research carefully creates a curated dataset and uses the BERT Transformer to extract features along with the KNN algorithm to group them together. Picking this grouping method is very important because it has a big effect on how FQDNs are put into groups. The resulting silhouette scores show us how distinct the clusters are. Greater silhouette scores, which signify well defined clusters, are consistent with the grouping technique's possible effectiveness in minimizing latency and maximizing communication paths. Higher silhouette scores mean that the groups are more clearly defined. The Agglomerative clustering with K=2 has a high silhouette score of 0.2930, which means that the areas are clear and well-separated. The best way to communicate might be through these well-organized clusters, especially if they are set up based on operational or physical closeness.

Lower silhouette scores for Agglomeration with K=5 and KMeans with K=5 mean that there are fewer clear groups. But the optimization goals depend on how the number of K is interpreted and the special features of DNS or FQDN data. In some situations, having more clusters (like K=7 or K=5) might give big differences that can be used to cut down on latency and find the best transmission paths. The study also acknowledges DBSCAN's density-based approach, suggesting that it could be useful for finding places with a lot of FQDN links. It might help to get lower latency if these dense areas line up with good communication routes.

The future of the study into improving MEL recognition in Osmotic Computing is broad, with many possible ways to keep improving and coming up with new ideas. As more research is done, clustering algorithms could be improved and made more efficient. Other methods could also be looked at besides the present integration with BERT, such as advanced NLP models like GPT or RoBERTa. Because Osmotic Computing environments are always changing, this is an interesting area to study further. Future work should focus on making the suggested method work with real-time changes in network conditions, resource access, and computing settings. Adding edge computing to the framework, looking into better privacy and security measures, and addressing worries about scalability will be very important for using it in the real world. In the future, more work might focus on putting the method to use in real-world large-scale computer situations, working with partners in the industry to make sure it works. The effects on the user experience, working together with experts from different fields, and helping with standardization efforts in the Osmotic Computing group are also important areas to look into in the future.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016). TensorFlow: A system for Large-Scale machine learning, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, USENIX Association, Savannah, GA, pp. 265–283.
URL: *https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi*

Alaparthi, S. and Mishra, M. (2020). Bidirectional encoder representations from transformers (BERT): A sentiment analysis odyssey, *CoRR* **abs/2007.01127**.
URL: *https://arxiv.org/abs/2007.01127*

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks, *Science* **286**(5439): 509–512.
URL: *https://www.science.org/doi/abs/10.1126/science.286.5439.509*

Cui, H., Yang, J., Liu, Y., Zheng, Z. and Wu, K. (2014). Data mining-based dns log analysis, *Annals of Data Science* **1**(3–4): 311–323.

Dhillon, I. S., Mallela, S. and Kumar, R. (2003). A divisive information-theoretic feature clustering algorithm for text classification, *J. Mach. Learn. Res.* **3**: 1265–1287.
URL: *https://api.semanticscholar.org/CorpusID:7827065*

Galletta, A., Sicari, C., Celesti, A. and Villari, M. (2021). Oce-dns: an innovative osmotic computing enabled domain name system.

Gandomi, A. and Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics, *International Journal of Information Management* **35**: 137–144.

Le, F., Wertheimer, D., Calo, S. and Nahum, E. (2022). Norbert: Network representations through bert for network analysis and management.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013). Distributed representations of words and phrases and their compositionality.

Monath, N., Dubey, A., Guruganesh, G. P., Zaheer, M., Ahmed, A. M. E. H., McCallum, A., Mergen, G., Najork, M., Terzihan, M., Tjanaka, B., Wang, Y. and Wu, Y. (2021). Scalable hierarchical agglomerative clustering, *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, p. 1245–1255.

Puliafito, C., Mingozzi, E., Longo, F., Puliafito, A. and Rana, O. (2019). Fog computing for the internet of things: A survey, *ACM Trans. Internet Technol.* **19**(2).
URL: *https://doi.org/10.1145/3301443*

Puthal, D., Sahoo, B., Mishra, S. and Swain, S. (2015). Cloud computing features, issues and challenges: A big picture.

Su, M.-Y. and Su, K.-L. (2023). Bert-based approaches to identifying malicious urls, *Sensors* **23**(20).
URL: *https://www.mdpi.com/1424-8220/23/20/8499*

Xu, Q., Migault, D., Sénécal, S. and Francfort, S. (2011). K-means and adaptive k-means algorithms for clustering dns traffic, *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '11, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, p. 281–290.

Yi, S., Li, C. and Li, Q. (2015). A survey of fog computing: Concepts, applications and issues, *Proceedings of the 2015 Workshop on Mobile Big Data*, Mobidata '15, Association for Computing Machinery, New York, NY, USA, p. 37–42.
URL: *https://doi.org/10.1145/2757384.2757397*