**Pre-thesis -II Report**



Deep Learning based Predictive Analytics for Efficient Content Caching in Edge Network

Name and ID:

1. Dhruba Chakraborty -18101028
2. Maisha Hossain -18201184
3. Saraf Noor Khaled - 18141009
4. Mahima Rabbi - 18101563
5. Maria Khanom Oishi - 17301029

**Supervisor: Md. Golam Rabiul Alam**

**Co-supervisor: Arif Shakil**

**Date of Submission: 3rd June, 2021**

**Department of Computer Science and Engineering**

**BRAC University**

Deep Learning based Predictive Analytics for Efficient Content Caching in Edge Network

**Abstract**

Content centric network is a state-of-the-art networking architecture for content distribution and content caching. However, it is inefficient to cache every content in each network device. The modern edge computing technology opens the door for content caching in the edge of the network. However, still we have to decide which contents we should cache and which content we should replace from the cache. Deep learning based predictive analytics can play an important role in selecting contents for caching purposes. In this research, we will use LSTM based Recurrent Neural Network for predictive content caching at the edge of the network.

1. **Introduction**

Soon after the invention of the first computer ENIAC in 1946, one of the most significant lacking it had was networking. People could do many things with the computer. But, it was impossible to share their works with others who were miles away. From this hunger of sharing, people started to think about making a system by which they could share their works with others. From this consequence, in 1960 ARPANET (The Advanced Research Projects Agency Network) was built in order to create a network with thousands of computers. And, thus the journey of networking had started.

In the very first era of networking, it was just a connection between computers for sharing mostly research data or important files. Only some of the sophisticated researchers and high-level people got to have the benefit of networking. But, in modern times, the concept of networking has changed a lot. Nowadays, there are thousands of fields in networking. People from every stage in society get help from networking in their day to day life. In this context, content has become the most powerful weapon in the networking field. People use content to get their job done in their daily life. Starting from media streaming sites, social networking sites, online news portals and many others are spreading digital wellbeing to human beings through content.

Content centric networks are getting richer day by day with the help of thousands of content providing sites and its users. However, this wouldn't have been this rich, if it wouldn’t have been efficient. Efficiently caching the contents is so important in networking. Caching a content means fetching the content from the server. It might be any server all over the world. But, that might be problematic as the server from which the files are being cached, might be far away from the user. That’s where efficient content caching comes in handy. In efficient content caching, files get fetched from the closest server. As a result, lots of time gets saved.

However, there is a significant issue when deciding which content we should cache and which we should replace from the cache because of limited cache memory. We need to cache contents that are more important to the users. But, it is harder to decide which content is more important to the user. To make the purpose easier, we can use deep learning based predictive analytics. Predictive analytics can help us to decide which file to cache and which file to replace from the cache depending on its importance.

**1.1. Research Problem**

With the mass availability of devices like mobile phones, computers etc. the use of the internet is increasing rapidly day by day. And, content providing sites like YouTube, Netflix, Prime Video etc. are becoming so popular among the users. However, people want to stream their contents faster from the sites with less latency. If the requested files are available on the cache server, they are delivered to the users extremely faster. Which is why caching is necessary. Assume a [1] Netflix subscriber in London wants to stream the show House of Cards. To ensure fast access and minimum buffering time, Netflix copies the videos from their origin servers in Los Gatos, CA, to the caching server closest to London. Because of this, all subscribers in London can quickly access the show and avoid a transatlantic file transfer. However, it is impossible to keep every movie in the closest caching server of London because of space limitation. To save the space of the cache server, the not so popular movies are needed to be replaced from the cache server with new ones. As a result, there comes a decision between what movies to keep and what movies need to be replaced from the cache.

Therefore, a question might arise:

***“Which contents we should cache and which contents we should replace?”***

This research will answer the above question through the usage of Deep Learning based Predictive Analytics Algorithm (in our case, LSTM based RNN).

**1.2. Research Objectives**

We are going to build a system using deep learning based predictive analytics so that we can decide which contents need to be cached and which contents need to be replaced from the server. The contents that are trending should be kept in the cache and others should be replaced from the cache. The objectives of the research are:

1. To understand, what content caching is and how it works
2. Importance of efficient content caching and its mechanism
3. Importance of edge computing and edge network in efficient content caching
4. To develop a model for connection between predictive analytics and efficient content caching
5. To evaluate the model
6. **Literature Review**

As the blessings of modern technologies like mobile phones, tablets, computers etc. are becoming more affordable and easier to get, people are getting more and more used to these devices’ day by day. And, people are getting more comfortable with content providing sites like Netflix, YouTube, Prime Video and so on. And, the number of users is rapidly increasing day by day. According to [2] Statista, the number of Netflix users in 2020 is 195.15 million by Q3. However, in a recent article of [3] TNPS (The New Publishing Standard), in 2030 the number of Netflix users is expected to increase up to 500 million.

There might be one problem with the loading time of the contents that are far away from the user. To solve that issue, the concept of caching comes in handy. But, the amount of cache memory is limited. That’s why there is a trade-off between which content to cache and which to replace. To efficiently cache data, predictive analytics is so necessary.

**2.1. Efficient Content Caching**

Content caching is a performance optimization mechanism in which data is delivered from the closest servers for optimal application performance. According to [4] ‘interserver’, when a system accesses the website, the contents in that site will be provided by a nearby cache server rather than the original server which is remote. As a result, it will decrease the latency. However, it is impossible to cache each and every content in the cache server. That’s why efficient content caching is needed. In efficient content caching, most popular contents are cached and least important contents get replaced from the cache server. It reduces server traffic and the performance of the application gets improved.

**2.2. Edge Computing and Edge Network**

Edge networking is a distributed computing paradigm that brings computation and data storage as close to the point of request as possible in order to deliver low latency and save bandwidth.However, [5] edge computing is a modern technology on data center and cloud computing architectures to help create efficiencies. Edge computing is significantly important outside the cloud, at the edge of the network, and more significantly in applications where real-time data processing is required. Due to the proximity of the analytical resources to the end users, sophisticated analytical tools and Artificial Intelligence tools can run on the edge of the system. According to [6] ‘The Emergence of Computing’, this placement at the edge helps to increase operational efficiency and contributes many advantages to the system.

**2.3. Predictive Analytics**

Predictive analytics is the use of data, statistical algorithms and machine learning techniques to identify the likelihood of future outcomes based on historical data. The goal is to go beyond knowing what has happened to providing a best assessment of what will happen in the future [7]. Though predictive analytics has been around for decades, it's a technology whose time has come. More and more organizations are turning to predictive analytics to increase their bottom line and competitive advantage. According to PredictiveAnalyticsToday [8], it uses a number of data mining, predictive modeling and analytical techniques to bring together the management, information technology and modeling business process to make predictions about the future. The patterns found in historical and transactional data can be used to identify risk and opportunities for the future.

**2.4. Related Works**

This part aims to critically review previous relevant works in the field of Predictive Analytics in the context of Efficient Content Caching at edge networks. Observing different techniques used in different relevant research works, we found many challenges in efficiently caching the contents through prediction.

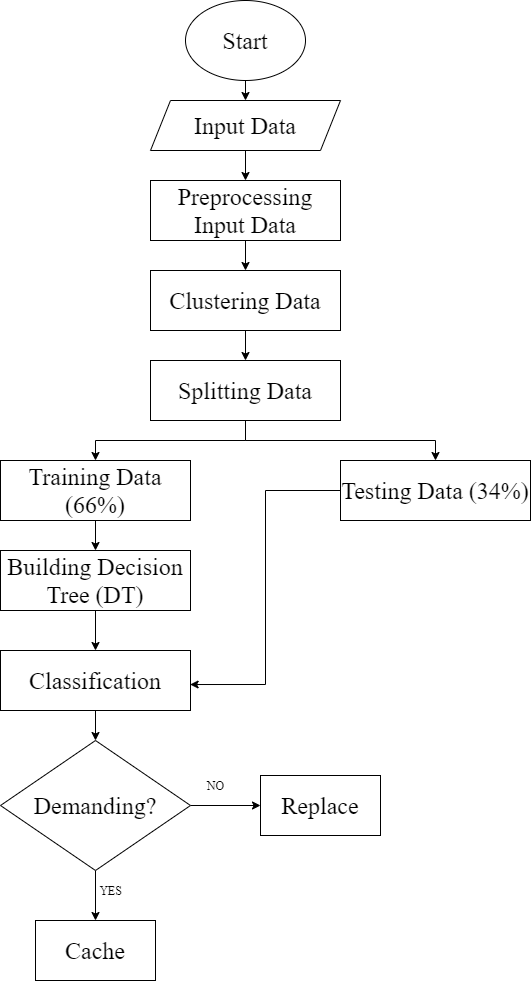
Content caching on the edge of the network is so important because if not cached, the data will be accessed by the user directly from the main server through the cloud. Which will increase the latency. According to [9], popular content and objects can be stored and served from edge locations, which are closer to the end users. This operation is also beneficial from the end user perspective since edge caching can dramatically reduce the overall latency to access the content and increase the sense of overall user experience.

Again, edge computing is another factor in terms of content caching. According to [10], using the cloud as a centralized server simply increases the frequency of communication between user devices, such as smartphones, tablets, wearables and gadgets, referred to as edge devices, and geographically distant cloud data centers. This is limiting for applications that require real-time response. Hence, there has been a need for looking ‘beyond the clouds’ towards the edge of the network, referred to as edge computing. Computing on edge nodes closer to application users could be exploited as a platform for application providers to improve their service. Although, the cache memory at the edge of the network is limited. So, we have to make a decision about what content to cache and what content to replace from the cache. That’s where deep learning based predictive analytics comes in useful.

Recurrent Neural Network (RNN) is significantly useful for solving the efficient content caching prediction problem because it not only utilizes the current state but also uses the previous state data using sequence. According to [9], Unlike the hidden neuron in FNN, the output of RNN depends on both the current output of the previous layer and the last hidden state. However, using RNN might not be appropriate in some cases as there might be data vanishing gradient problems. Recurrent Neural Networks work just fine when we are dealing with short-term dependencies[20]. To solve that issue, LSTM (Long Short-Term Memory) comes handy. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems[14]. According to [10], LSTM models are quite popular due to their special design property related to carefully avoiding vanishing and exploding gradient problems when building deep layer neural network models. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things[20].

1. **Methodology**

The aim of using predictive analytics for efficient content caching at the edge network is to cache the most popular contents at the edge of the network and thus decrease the latency. With a view to doing so, the model requires designing a process that takes data from the activity of the users as an input. Then it systematically processes the input data and outputs either of the two different results: ‘cache’ or ‘replace’. The below figure provides a high-level view of the model design:



**Figure 1**: *The flow chart of the predictive analytics model*

We are using LSTM based Recursive Neural Network (RNN) for solving our problem. We could use other Deep Learning based models for this work. But, unlike many other algorithms, LSTM based RNN remembers the previous sequence by keeping them in memory. As a result, the output gets more and more accurate day by day. The workflow will be done in the following stages:

1. **Input data:**

In this stage the program takes activity data from the users as input.

1. **Input data preprocessing:**

In this stage the input data gets formatted in such a way that LSTM can use it to process easily.

1. **Processing:**

In this stage the formatted input data gets clustered into groups. After clustering, the preprocessed input data is split into two groups; one group is used for training and building the decision tree, and the other group is used for testing the accuracy of the decision tree.

1. **Predictions:**

In this stage DT is used for prediction to decide whether to cache or replace from cache.

**3.1. Dataset Description**

In predictive analytics, the most important tool is data. To know what to cache and what not to cache, we need a lot of data based on the user's ratings. That is why we have chosen MovieLens dataset which consists of various datasets among which, we will primarily be using movies and ratings datasets. The movie dataset consists of movie id, title, genres and rating dataset consists of userid, movie id, rating and timestamp. It contains 27753444 ratings and 1108997 tag applications across 58098 movies. These data were created by 283228 users between January 09, 1995 and September 26, 2018. This dataset was generated on September 26, 2018 [11]. This dataset can be downloaded from [12].

**3.2. Data Pre-Processing**

Data preprocessing is a data mining technique to turn the raw data gathered from diverse sources into cleaner information that’s more suitable for work. In other words, it’s a preliminary step that takes all of the available information to organize it, sort it, and merge it[13].

* **Dataset clean:** Not all the data of a dataset are necessary for each and every research. For that reason, dataset cleanup is necessary for preparing the data for pre-processing. Data cleaning is required for smooth noisy data and standardizing the data. By cleaning up the dataset movies, we are categorizing the genres into integer values and adding another field called release date which gets derived from the title field. In the case of rating dataset, we are sorting the dataset by userid keyword. Also, a field called daily (seconds a day).

|  |  |  |
| --- | --- | --- |
| movieId | title | genres |
| 1 | Toy Story (1995) | Adventure|Animation|Children|Comedy|Fantasy |
| 2 | Jumanji (1995) | Adventure|Children|Fantasy |
| 3 | Grumpier Old Men (1995) | Comedy|Romance |
| 4 | Waiting to Exhale (1995) | Comedy|Drama|Romance |
| 5 | Father of the Bride Part II (1995) | Comedy |

|  |  |  |  |
| --- | --- | --- | --- |
| userId | movieId | rating | timestamp |
| 1 | 307 | 3.5 | 1260000000 |
| 1 | 481 | 3.5 | 1260000000 |
| 1 | 1091 | 1.5 | 1260000000 |
| 1 | 1257 | 4.5 | 1260000000 |
| 1 | 1449 | 4.5 | 1260000000 |

*fig: Before Data Cleanup*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | movieId | title | genres | releaseDate |
| 0 | 1 | Toy Story (1995) | 1.00 | 1995 |
| 1 | 2 | Jumanji (1995) | 2.00 | 1995 |
| 2 | 3 | Grumpier Old Men (1995) | 3.00 | 1995 |
| 3 | 4 | Waiting to Exhale (1995) | 4.00 | 1995 |
| 4 | 5 | Father of the Bride Part II (1995) | 5.00 | 1995 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| timestamp | userId | movieId | rating | daily |
| 23237827 | 237556 | 21 | 3 | 9140 |
| 5510411 | 56769 | 1176 | 4 | 9140 |
| 23237876 | 237556 | 1079 | 3 | 9140 |
| 23237833 | 237556 | 47 | 5 | 9140 |
| 23096009 | 236139 | 28 | 5 | 9524 |

*fig: After Data Cleanup*

* Then we are joining the datasets (movies, rating) that are coming from our previous step, dataset cleanup.

* After joining the datasets, we are passing the joined.csv into our pre-processing algorithm. At the very beginning the timestamps get converted into category. The main purpose of pre-procesing the dataset is to get the input dataset labeled and make the dataset prepared for our research purpose.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | timestamp | userId | movieId | rating | daily | tstamp\_hour | tstamp\_day | tstamp\_year | genres | releaseDate |
| 0 | 23237827 | 237556 | 21 | 3 | 9140 | 6455 | 269 | 1 | 17 | 1995 |
| 1 | 5510411 | 56769 | 1176 | 4 | 9140 | 1531 | 64 | 1 | 93 | 1991 |
| 2 | 23237876 | 237556 | 1079 | 3 | 9140 | 6455 | 269 | 1 | 38 | 1988 |
| 3 | 23237833 | 237556 | 47 | 5 | 9140 | 6455 | 269 | 1 | 30 | 1995 |
| 4 | 23096009 | 236139 | 28 | 5 | 9524 | 6416 | 268 | 1 | 15 | 1995 |
| 5 | 2619774 | 26999 | 60 | 4 | 9524 | 728 | 31 | 1 | 2 | 1995 |
| 6 | 23096013 | 236139 | 58 | 5 | 9524 | 6416 | 268 | 1 | 4 | 1994 |

*fig: Before pre-processing*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | timestamp | userId | movieId | rating | daily | tstamp\_hour | tstamp\_day | tstamp\_year | genres | releaseDate | label | label\_nom |
| 0 | 23237827 | 237556 | 21 | 3 | 9140 | 6455 | 269 | 1 | 17 | 1995 | 18 | 0.005144 |
| 2 | 23237876 | 237556 | 1079 | 3 | 9140 | 6455 | 269 | 1 | 38 | 1988 | 1 | 0.000286 |
| 3 | 23237833 | 237556 | 47 | 5 | 9140 | 6455 | 269 | 1 | 30 | 1995 | 23 | 0.006573 |
| 51498 | 23193844 | 237134 | 21 | 5 | 9609 | 6443 | 269 | 1 | 17 | 1995 | 18 | 0.005144 |
| 51499 | 23193845 | 237134 | 150 | 5 | 9609 | 6443 | 269 | 1 | 58 | 1995 | 40 | 0.011432 |

*fig: after pre-processing*

* Then we are sorting our pre-processed dataset in ascending order based on timestamp, userId, movieId.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | timestamp | userId | movieId | rating | daily | tstamp\_hour | tstamp\_day | tstamp\_year | genres | releaseDate | label | label\_nom |
| 0 | 23237827 | 237556 | 21 | 3 | 9140 | 6455 | 269 | 1 | 17 | 1995 | 18 | 0.005144 |
| 2 | 23237876 | 237556 | 1079 | 3 | 9140 | 6455 | 269 | 1 | 38 | 1988 | 1 | 0.000286 |
| 3 | 23237833 | 237556 | 47 | 5 | 9140 | 6455 | 269 | 1 | 30 | 1995 | 23 | 0.006573 |
| 51498 | 23193844 | 237134 | 21 | 5 | 9609 | 6443 | 269 | 1 | 17 | 1995 | 18 | 0.005144 |
| 51499 | 23193845 | 237134 | 150 | 5 | 9609 | 6443 | 269 | 1 | 58 | 1995 | 40 | 0.011432 |

*fig: before sorting*

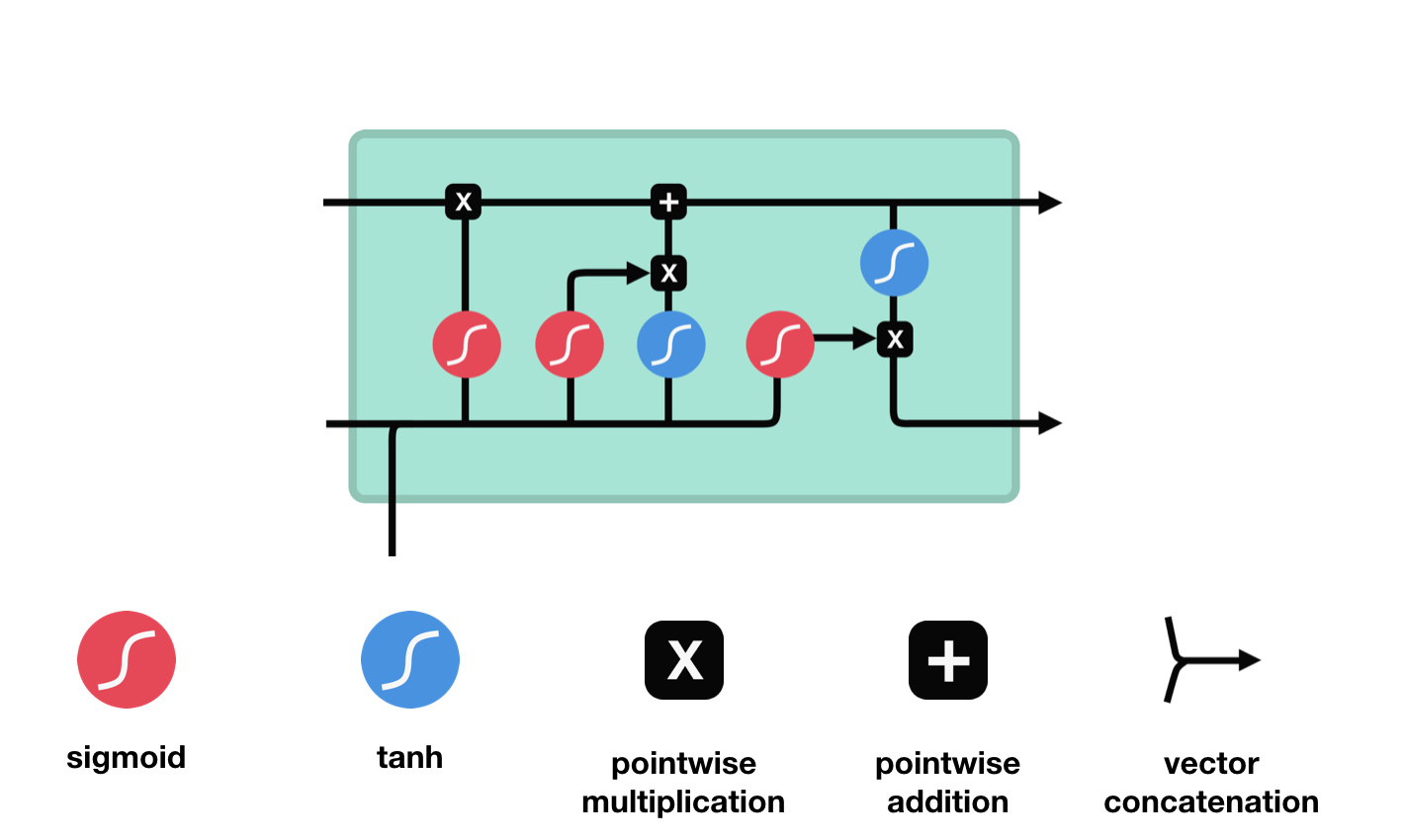
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | timestamp | userId | movieId | rating | daily | tstamp\_hour | tstamp\_day | tstamp\_year | genres | releaseDate | label | label\_nom |
| 64 | 23184175 | 237014 | 1 | 3 | 9618 | 6441 | 269 | 1 | 1 | 1995 | 14 | 0.004001 |
| 65 | 23184176 | 237014 | 4 | 4 | 9618 | 6441 | 269 | 1 | 4 | 1995 | 5 | 0.001429 |
| 66 | 23184177 | 237014 | 10 | 3 | 9618 | 6441 | 269 | 1 | 9 | 1995 | 28 | 0.008002 |
| 63 | 23184178 | 237014 | 11 | 5 | 9618 | 6441 | 269 | 1 | 4 | 1995 | 14 | 0.004001 |
| 62 | 23184179 | 237014 | 19 | 1 | 9618 | 6441 | 269 | 1 | 5 | 1995 | 13 | 0.003715 |

*fig: after sorting*

**3.3 Long Short-Term Memory (LSTM)**

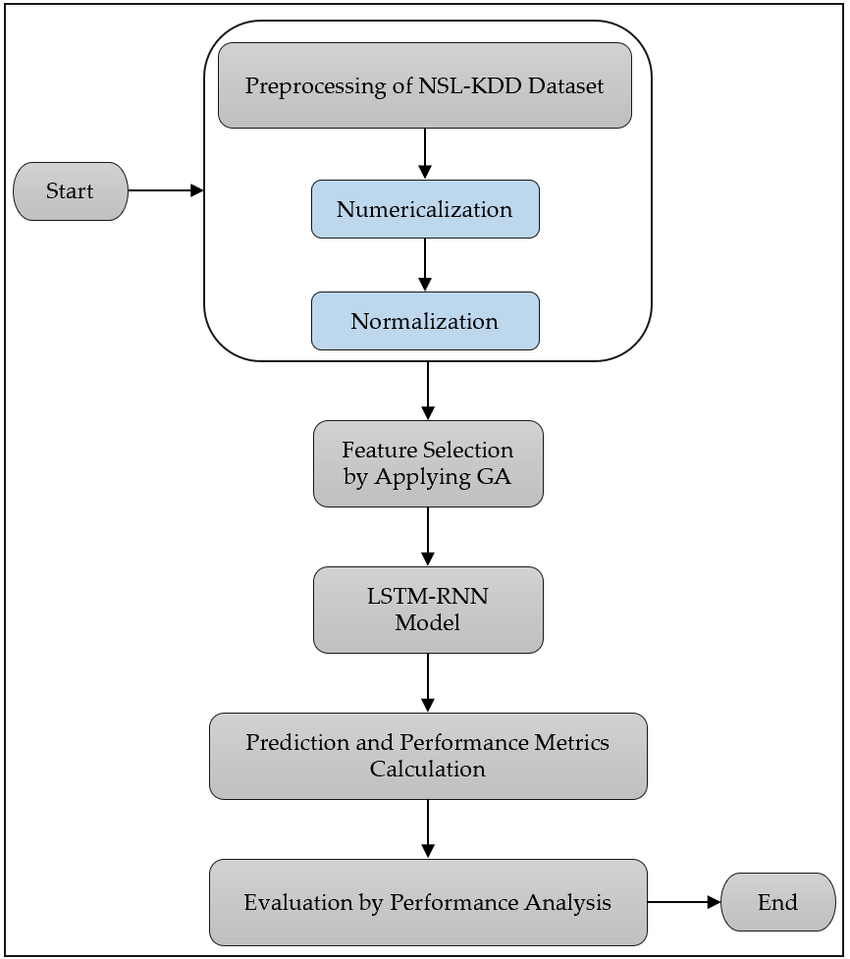
Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems[14]. LSTM networks are well-suited to classifying, processing and making predictions based on time series data[15]. LSTM is an RNN architecture specifically designed to address the vanishing gradient problem[14]. LSTM works tremendously well on a large variety of problems, and are now widely used[16].

Here is the structure of the Long Short-Term Memory(LSTM) unit which shows its workflow. A **LSTM** unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

**

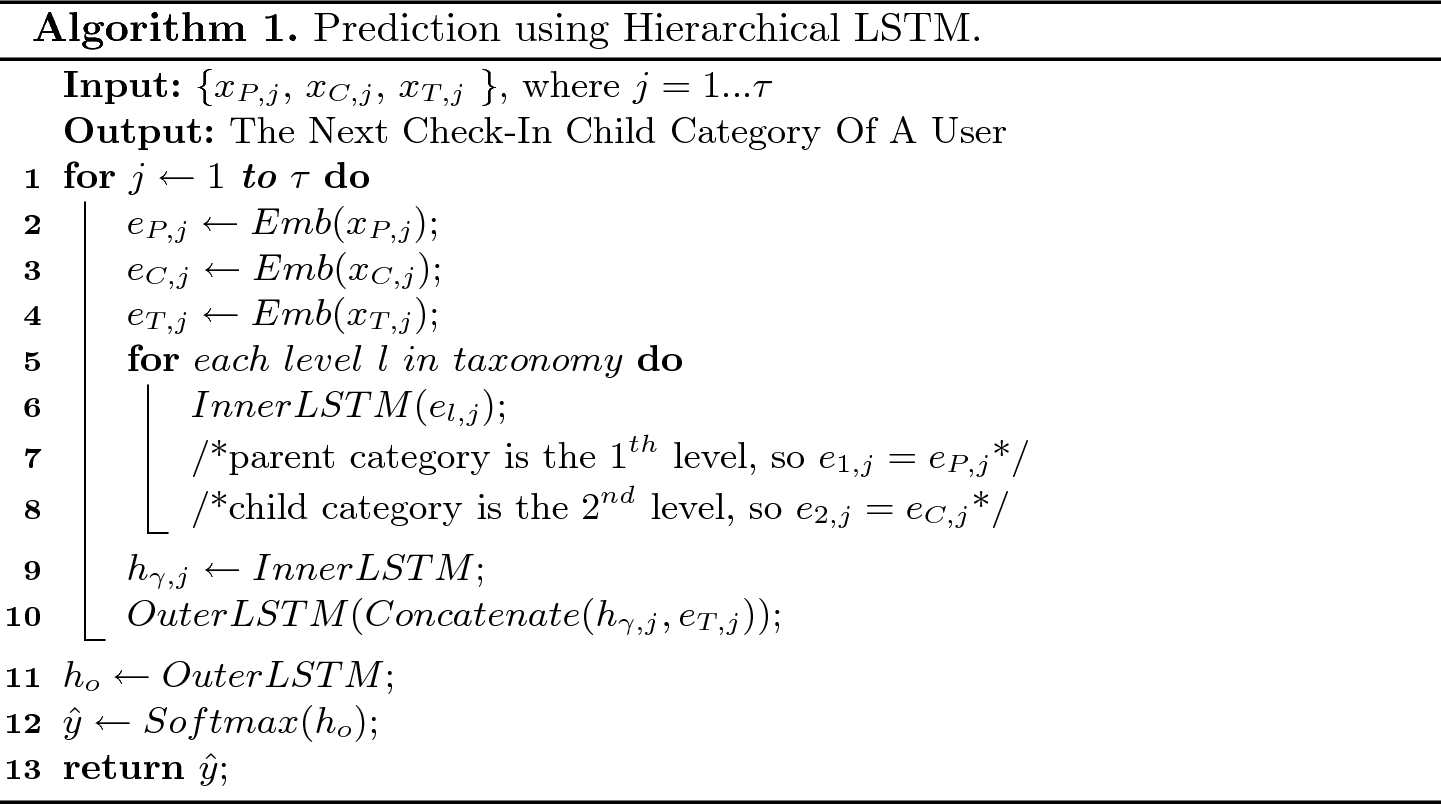
*fig: Structure of a LSTM Unit [17]*

Now, flowchart of **LSTM** has been given here to illustrate the work process and its steps:

**

*fig: Flowchart of LSTM [18]*

Finally, Algorithm of **LSTM** has been given here to illustrate the work process and its steps:

**

*fig: LSTM algorithm [19]*

**IV. Implementation and Result**

This section describes the implementation of the proposed model for predicting contents for the users in the edge of the network. This model was implemented and tested using Jupyter Notebook. The implementation of the model consists of dataset collection, input data pre-processing and testing. Among them, we have already described the first two parts previously. Now, we are only describing the testing part.

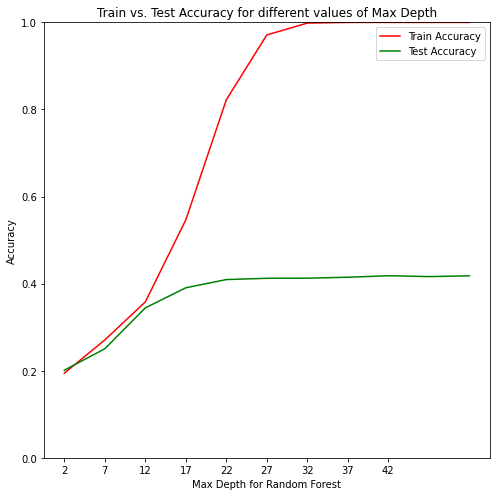
This section also delivers the result of running the implementation of the proposed model for predicting contents. Jupyter Notebook is used to run the test file. Jupyter Notebook is a powerful tool for running python codes. We could have used languages like Java or C. However, python is much more efficient and much less time consuming compared to those languages. Also, most of the machine learning libraries are easily accessible compared to Java or C. That is why we have chosen python as our primary programming language.

**4.1. Implementation**

The proposed model consists of five files. The files are described in table below:

|  |  |
| --- | --- |
| **File Name** | **Description** |
| maincsv.ipynb | Cleans up the dataset and categorises genres into integers. |
| join\_dataset.ipynb | Joins two datasets (movies.csv and ratings.csv) |
| preprocessing.ipynb | labels up the data and prepares for applying prediction based algorithms. |
| sort.ipynb | Ascendingly sorts the preprocessed dataset based on timestamp, userId, movieId. |
| prediction.ipynb | Applies LSTM on the dataset and provides predicted data. |

In prediction.ipynb we have taken the processed dataset as input. Then we have chosen 60% as the training dataset and 40% as the testing dataset. Then we applied a random forest algorithm to visualize the accuracy for different regularization parameters.



*fig: Accuracy for different values of max depth (train vs. test)*

**V. Conclusion**

Predictive analytics is highly important for efficiently caching the contents at the edge of the network. Predictive analytics allows organizations to become proactive, forward-looking, anticipating outcomes and behaviors based upon the data and not on guesses or assumptions. So is applicable for content caching. Predictive analytics helps the content providers to cache the most popular contents at the edge of the networks so that users can access them faster with lower latency.

**References**

1. [What is Content Caching? - Articles for Developers Building High Performance Systems (stackpath.com)](https://blog.stackpath.com/glossary-content-caching/)
2. [• Netflix: number of subscribers worldwide 2020 | Statista](https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/)
3. [Netflix targets 500 million subscribers by 2030. Rolls out mobile-only subscription option for Nigeria’s 126 million internet users. Publishers look the other way - The New Publishing Standard](https://thenewpublishingstandard.com/2020/10/05/netflix-targets-500-million-subscribers-by-2030-rolls-out-mobile-only-subscription-option-for-nigerias-126-million-internet-users-publishers-look-the-other-way/#:~:text=A%20half%20billion%20subscribers%20by,from%20Digital%20TV%20Research%20shows.)
4. [What is Content Caching? - Interserver Tips](https://www.interserver.net/tips/kb/what-is-content-caching/)
5. [Archive | Design Automation Conference (dac.com)](http://www2.dac.com/events/videoarchive.aspx?confid=170&filter=keynote&id=170-103--0&#video)
6. [The Emergence of Edge Computing - IEEE Journals & Magazine](https://ieeexplore.ieee.org/abstract/document/7807196) [Predictive Analytics: What it is and why it matters | SAS](https://www.sas.com/en_us/insights/analytics/predictive-analytics.html)
7. [What is Predictive Analytics ? (predictiveanalyticstoday.com)](https://www.predictiveanalyticstoday.com/what-is-predictive-analytics/)
8. [[2008.07235] A Survey of Deep Learning for Data Caching in Edge Network (arxiv.org)](https://arxiv.org/abs/2008.07235)
9. [(PDF) Challenges and Opportunities in Edge Computing (researchgate.net)](https://www.researchgate.net/publication/307888359_Challenges_and_Opportunities_in_Edge_Computing)
10. [DeepCache | Proceedings of the 2018 Workshop on Network Meets AI & ML (acm.org)](https://dl.acm.org/doi/10.1145/3229543.3229555)
11. [MovieLens Latest Full | Kaggle](https://www.kaggle.com/grouplens/movielens-latest-full/version/1)
12. <http://files.grouplens.org/datasets/movielens/ml-latest.zip>
13. [Data Preprocessing: what is it and why is important - CEOWORLD magazine](https://ceoworld.biz/2019/12/13/data-preprocessing-what-is-it-and-why-is-important/)
14. [A Gentle Introduction to Long Short-Term Memory Networks by the Experts (machinelearningmastery.com)](https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/)
15. [Long Short Term Memory (LSTM) (opengenus.org)](https://iq.opengenus.org/long-short-term-memory-lstm/)
16. [Understanding LSTM Networks -- colah's blog](https://colah.github.io/posts/2015-08-Understanding-LSTMs/)
17. [Illustrated Guide to LSTM’s and GRU’s: A step by step explanation | by Michael Phi | Towards Data Science](https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21)
18. [Flow-chart of the LSTM-RNN classifier with GA. | Download Scientific Diagram (researchgate.net)](https://www.researchgate.net/figure/Flow-chart-of-the-LSTM-RNN-classifier-with-GA_fig2_341126962)
19. [Hierarchical LSTM: Modeling Temporal Dynamics and Taxonomy in Location-Based Mobile Check-Ins | SpringerLink](https://link.springer.com/chapter/10.1007/978-3-030-16145-3_17)
20. [Long Short Term Memory | Architecture Of LSTM (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/)