Edge computing with IoT and Cloud model

Paper Summary

The amount of data generated from various devices is enormous in the present times. To aid that amount of generation, the number of connected devices or so-called IoT devices has seen a tremendous increase in the past few years. Cloud computing technology is used to facilitate the transmission and collection of data from these IoT devices. It acts as a collector of real-time data as well as a provider of services built on that data. And one such service is data analytics. It is a primary service offered by many cloud computing systems since those huge data are put to good use for generating invaluable insights, and key decisions. Thus boosting the productivity and services of the company.

Edge computing is used to send the data from IoT devices to cloud servers and receive insights from them efficiently. And to suffice this Deep learning(DL) model called Auto Encoder(AE) is used to retrieve only the useful features from data which is then reduced to its encoded form and sent to the cloud where the Machine learning model is applied either directly on the data or the decoded form of the data which is the original data with some loss from the reconstruction.

Even though edge does computations it is merely enough to do the computation that an ML model needs. So, all it does is reduce the network traffic and redundancy in data that is sent to the cloud. So AE is a part of the DL bottleneck Neural network that encodes input and produces output. This is trained separately and the trained model is placed on the edge which then performs encoding for data generated from the IoT source. The decoding part is trained and placed on the cloud which reconstructs the original data and passes it to an ML model for data analytics.

The edge cloud ML system model consists of three stages data pre-processing, data reduction, and cloud ML. The data Pre-processing stage employs the normalization method to treat the outliers as well as the difference in values between variables. It uses the standardization method, i.e standard score $z = (x - \mu)/\sigma$ to normalize the data points. It performs better than the min-max scaler because it removes the outlier and the data point distribution is free from outlier influence. The next method is optional which is the sliding window technique. It helps to find the patterns and periodical behavior in the data points. So it uses a matrix to represent the sliding pattern for each sliding step and thus tries to reduce the number of features by identifying the dependencies among them after collecting the required number of sliding patterns(matrices).

Once data is preprocessed it is fed into the data reduction phase which consists of two types. One is reversible reduction, i.e it can be reconstructed to the original data on the cloud. Another type is the non-reversible reduction which cannot be reconstructed to its original form in the cloud. reversible reduction is carried out with PCA that reduces the dimensionality hence reducing the features of the data by AE. Which can be reconstructed to its original form by the help of eigenvectors that were generated in the edge. ANd there are three methods of selecting the data for reduction, first one is data from all the sensors, and reduction is carried at once to the whole set. Next data is grouped location-wise of the sensors and their respective edge apply PCA and reduce the data. The other is grouping the data by the similarity of sensors and reducing the data for similar groups. Non-reversible reduction on the other hand is done by finding the magnitude of the reduced data and only the magnitude is sent to the cloud. The dimensionality is reduced to 3:1 ratio repeatedly by $d = \sqrt{x^2 + y^2 + z^2}$, where d is the magnitude and x,y,z are the features. The ML model in the cloud then directly use the magnitude data.

The last phase is the cloud ML which is sending the reduced data from edge to cloud. In the cloud ML model is deployed which acts either directly on the reduced data, which is lower in size and requires much less computational resources, or reconstructs the original data if it is produced by the reversible technique and then acts on it. But it takes way more computational resources and time for the model than directly training on reduced data. And by this way the model had reduced the network traffic as well as the amount of data that's transferred than a traditional IoT- cloud model through the use of edge along with the source of data generation. This model is highly effective for applications that require fast response time and those with limited bandwidth due to the fact that computation is carried out near the sensors.

Key ideas from author

- Edge computing is used to reduce the network traffic between sensors and cloud servers. Thus, increasing the efficiency as well as the robustness of the system.
- It employs AutoEncoder(AE) which is a trained deep learning model of a bottleneck neural network that encodes the data from IoT sensors at the edge side and decodes it to reconstruct original data at the cloud side.
- The edge-cloud ml model consists of three phases- data preprocessing, data reduction, and cloud ml.
- data pre-processing has two methods: Normalization using standard score and window sliding technique to reduce the features.
- data reduction phase consists of two types. One is reversible reduction, i.e it can be reconstructed to the original data on the cloud. Another type is

the non-reversible reduction which cannot be reconstructed to its original form in the cloud.

- reversible reduction is carried out with PCA that reduces the dimensionality hence reducing the features of the data by AE. Which can be reconstructed to its original form by the help of eigenvectors that were generated in the edge. ANd there are three methods of selecting the data for reduction, first one is data from all the sensors and reduction is carried at once to the whole set. Next data is grouped location-wise of the sensors and their respective edge apply PCA and reduce the data. The other is grouping the data by the similarity of sensors and reducing the data for similar groups.
- non-reversible reduction on the other hand is done by finding the magnitude of the reduced data and only the magnitude is sent to the cloud. The ML model in the cloud then directly use the magnitude data.
- The reduced form of data is sent to the cloud where it hosts an ML model that acts on the data directly or reconstructs it and acts.

My views

- Introduction of edge computing into the IoT and Cloud interface is an efficient and robust mechanism. It indeed reduces the network traffic as well as the data bandwidth that is shared to the cloud.
- AE reduces the errors in the data, And functions suitably well on the edge.
- Data is pre-processed using the standard score and window sampling which are quite powerful and efficient.
- The data is reduced in the edge through PCA which is a good data-reducing technique but I would like to apply some of the filter methods for feature reduction which is quite computationally less expensive and increases performance.
- Then reduced data is sent to the cloud ML which acts on the data and generates much-needed insights.

Agreements, Pitfalls, and Fallacies

VIEWS	INFERENCES
Edge computing is used to reduce the network traffic between sensors and cloud servers. Thus, increasing the efficiency as well as the robustness of the system.	I completely agree with the fact that the use of edges in the IoT- cloud interface will reduce the amount of data that is transferred to the cloud and also ensures quality and resourcefulness in the data that is passed which in turn reduces the data cleaning and outlier treatment overhead at the cloud side
AE is used on the edge to serve its purpose of reducing the data size that uses a trained bottleneck neural network.	I agree with that fact since the deep learning model is the present go-to technique when the variety and volume of data handled are versatile.
data pre-processing has two methods: Normalization using standard score and window sliding technique to reduce the features	I agree with that approach since standardization removes outliers and will be from the outlier influence while training,
reversible reduction is carried out with PCA	I partially agree with the PCA method but I think that indeed demands quite an amount of computational resources and time which will demand the edge to be computationally powerful. So, some easier techniques like Filter models such as fischer score can be used,

Submission by:

Dhev Mugunddhan A
21011101040
AI & DS - A
paper: Cloud Computing for Internet of Things Data Analytics Embedding Intelligence in the Edge With Deep Learning by Ananda Mohon Ghosh and Katarina Grolinger published by IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 17, NO. 3, MARCH 2021