

# Pengujian Long-Short Term Memory (LSTM) Pada Prediksi Trafik Lalu Lintas Menggunakan Multi Server

Riesa Krisna Astuti Sakir<sup>1)</sup>

<sup>1</sup> Jurusan Teknik Elektro, Politeknik Negeri Ujung Pandang  
riesakrisna@poliupg.ac.id



## Abstract

*This study presents a test of the long short term memory (LSTM) algorithm on traffic prediction with multi edge server and cloud server architectures. IoT sensors located on the roadside such as cameras and location data on each driver are used and stored in the data center. When a driver sends a travel time request to a nearby edge server, traffic predictions will be made on the edge server or cloud server. Server selection is made based on the destination location of the driver's request. If the destination is in the edge server area, traffic predictions are made on the edge server. However, if the destination is in the cloud server area, traffic predictions are made on the cloud server. Then to predict traffic is done with LSTM. following modeling is made with a density of 128 and a density of 256. By learning from previous traffic, LSTM with a greater density gets a proportion of errors, namely RMSE 10.78%, MAE 8.24%, and MAPE 19.87%.*

**KeyWords:** *LSTM, Multi Edge Server and Cloud Server, Traffic Light, Travel Time*

## Abstrak

*Penelitian ini menyajikan pengujian algoritma long short term memory (LSTM) pada prediksi trafik lalu lintas dengan arsitektur multi edge server dan cloud server. Sensor IoT yang terletak di tepi jalan seperti kamera dan data lokasi pada setiap pengendara digunakan dan disimpan pada data center. Ketika pengendara mengirimkan permintaan waktu tempuh kepada edge server disekitarnya, maka prediksi trafik akan dilakukan di edge server atau cloud server. Pemilihan server dilakukan berdasarkan lokasi tujuan permintaan pengendara. Jika tujuan berada pada area edge server maka prediksi trafik dilakukan pada edge server tersebut. Tetapi, jika tujuan berada pada area cloud server maka prediksi trafik dilakukan pada cloud server. Kemudian, untuk prediksi trafik lalu lintas dilakukan dengan LSTM. berikut pemodelan yang dibuat dengan dense 128 dan dense 256 Dengan pembelajaran pada trafik sebelumnya, LSTM dengan dense yang lebih besar mendapatkan persentase kesalahan yaitu RMSE 10.78%, MAE 8.24%, dan MAPE 19.87%.*

**Kata Kunci:** *LSTM, Multi Edge Server dan Cloud Server, Trafik Lalu Lintas, Waktu Tempuh.*

## I. PENDAHULUAN

Konsep *smart city* mengacu pada penggunaan berbagai internet of thing (IoT) ensor di daerah perkotaan [1]. Sensor IoT terletak di pinggir jalan, di area lampu lalu lintas, di area parkir, dan persimpangan untuk mengukur perubahan informasi kendaraan, jalan, dan pejalan kaki. Informasi tersebut kemudian dimanfaatkan oleh pusat manajemen lalu lintas untuk meningkatkan sistem transportasi cerdas (ITS) [2]. Pengontrolan lalu lintas dan sistem manajemen kendaraan, dimaksudkan untuk menghindari terjadinya masalah yang sering terjadi seperti kemacetan, adanya kendaraan darurat (*ambulance*), dan kecelekaan). Kontrol sinyal lalu lintas dan

prediksi arus lalu lintas adalah cara alternatif untuk mengontrol lalu lintas kendaraan.

Salah satu prediksi lalu lintas yang biasa digunakan pengendara adalah sistem navigasi *google maps*, dimana menggunakan posisi *cell phones*. Jika pada suatu titik banyak terdeteksi *cell phones* maka titik tersebut terjadi kemacetan dan sebaliknya [3]. Kemudian memperkirakan waktu tempuh untuk sampai ke tujuan. Dalam memperkirakan waktu terdapat beberapa algoritma yang digunakan seperti random forest algorithm, road link dynamic dividing algorithm, and speed estimation algorithm.

Selain data saat ini seperti posisi *cell phones*, pada beberapa penelitian mengenai prediksi juga memanfaatkan data sebelumnya

sebagai pembelajaran. Algoritma *long-short term memory* (LSTM) merupakan salah satu diantara *machine learning* yang melakukan pembelajaran pada data sebelumnya. LSTM termasuk bagian *Recurrent Neural Network* (RNN). Akurasi prediksi menggunakan LSTM cukup baik untuk beberapa implementasi. Seperti prediksi nilai indikator pencemaran udara [4], prediksi curah hujan [5], prediksi trafik lalu lintas [6], dan ramalan *web traffic* [7].

Selanjutnya, melihat perangkat elektronik yang terletak di area jalan seperti kamera, juga dapat dimanfaatkan sebagai sensor dalam mendeteksi kemacetan. Serta pemanfaatan edge server, dimana jaraknya lebih dekat ke pengendara dapat mempercepat pengambilan keputusan prediksi waktu tempuh atau estimasi trafik lalu lintas. Penggunaan edge server mendukung pengembangan aplikasi intelligent transportation system [8].

Edge server termasuk ke bagian edge computing yang hadir untuk mengatasi permasalahan latency tinggi oleh cloud computing [9]. Selain itu, edge computing juga mendukung pemantauan secara real-time. Namun, memiliki kapasitas penyimpanan yang kecil dibandingkan cloud computing. Sehingga dibutuhkan koneksi yang baik antara edge server dan cloud server dalam memantau trafik lalu lintas serta mengestimasi trafik untuk menghindari kemacetan. Cloud server dimaksudkan untuk mengatasi permasalahan jika pengendara meminta tujuan yang berada di luar area edge server, sedangkan edge server mengatasi permasalahan untuk pengendara yang berada pada areanya.

Berdasarkan penjelasan diatas, maka penelitian ini akan mengimplementasikan LSTM untuk estimasi trafik dengan menempatkan edge server dan cloud server sebagai bentuk kontribusi pada pengembangan aplikasi ITS. Secara rinci penelitian ini melakukan:

- Membuat alur pemrosesan data untuk melakukan estimasi trafik pada edge server atau cloud server
- Mengestimasi trafik menggunakan LSTM menggunakan dua kondisi, sibuk dan non-sibuk.

Selanjutnya penelitian ini disusun sebagai berikut. Bagian 2 mengkaji literatur mengenai LSTM dan perbandingan cloud computing, fog computing, dan edge computing. Bagian 3 menjelaskan metode penelitian. Di Bagian 4

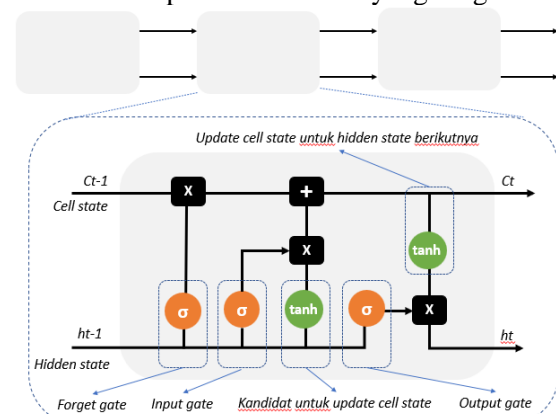
menjabarkan hasil dan pembahasan. Kemudian, di Bagian 5 merangkum penelitian ini dengan kesimpulan.

## II. KAJIAN LITERATUR

### A. Algoritma LSTM

LSTM merupakan pengembangan dari RNN yang digunakan memprediksi trafik lalu lintas untuk meningkatkan performansi dari estimasi waktu tempuh. Secara umum, struktur algoritma LSTM terdiri dari neural network dan blok memori yang disebut sebagai *cell* yang digunakan untuk menyimpan masukan yang diambil dalam waktu tertentu seperti yang ditunjukkan pada Gambar 2. State dari *cell* dan *hidden state* akan diteruskan *cell* berikutnya. Pada *cell* terdapat tiga gate yaitu *input gate*, *output gate*, dan *forget gate*.

*Input gate* berfungsi menambah informasi untuk masuk ke *cell*. Informasi yang masuk diatur menggunakan fungsi *sigmoid* dan melakukan *filter* untuk menyaring nilai yang akan disimpan. Dengan fungsi *tanh*, sebuah vektor dibuat untuk memberikan nilai dari proses *sigmoid* dan *filter*. Selanjutnya dikalikan untuk mendapatkan informasi yang berguna.



Gambar 2. Gambaran umum arsitektur LSTM

*Output gate* bertugas untuk mengekstrak informasi dari *cell state*. Diawali dengan pembangkitan vektor menggunakan fungsi *tanh* pada *cell*, informasi kemudian diatur menggunakan fungsi *sigmoid* dan menyaring nilai-nilai yang akan tersimpan. Setelah itu, nilai vektor dan nilai yang diatur dikalikan dan dikirim sebagai *output* dan input ke *cell* berikutnya.

*Forget gate* digunakan untuk menghapus informasi yang tidak lagi digunakan dengan cara melakukan evaluasi *output* biner dari *input cell* dan *output cell* sebelumnya. Kemudian dikalikan dengan matriks bobot dan

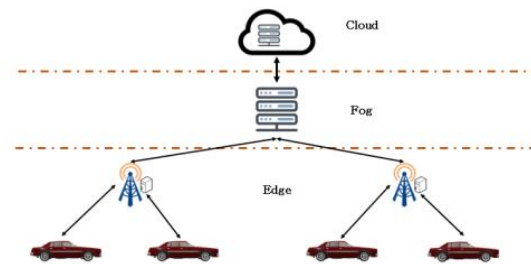
ditambahkan nilai bias. Nilai yang didapat akan menghasilkan *output* biner dengan melalui fungsi aktivasi. Jika *output* bernilai 0, maka informasi sudah tidak berguna dan akan dihapus. Sedangkan, jika *output* bernilai 1 maka informasi disimpan untuk penggunaan di masa mendatang.

### B. Perbandingan Cloud Computing, Fog Computing, dan Edge Computing

Tabel 1. Perbandingan cloud computing, fog computing, dan edge computing.

	Cloud Computing	Fog Computing	Edge Computing
Latency	high	moderate	Low
Delay jitter	high	low	Very low
Lokasi service	Dalam internet	Antara cloud dan edge server	Pada tepi local network
Security	Sulit ditemukan	Mudah ditemukan	Mudah ditemukan
Location awareness	Tidak ada	Ada	Ada
Jarak antara client dan server	Multiple hops	Multiple hops	One hop
Number of server nodes	Sedikit	Sedang	Besar
Real-time interaction	Mendukung	Mendukung	Mendukung
Support for mobility	Limited	Mendukung	Mendukung

Pemantauan trafik lalu lintas secara real-time, kurang efektif jika menggunakan cloud server, karena jarak yang jauh dari sensor IoT dan pengendara. Kemudian fog computing dan edge computing mengatasi masalah jarak yang jauh untuk pemantauan secara real-time dengan menggunakan edge server. Keduanya terletak di tepi jaringan seperti stasiun pangkalan atau di dekat perangkat. Berdasarkan uraian tersebut, posisi cloud server dan edge server dapat ditampilkan di Gambar 1. Tingkat Layanan Komputasi Kemudian, detail layanan mereka perbedaan tercantum dalam Tabel 1 Perbandingan cloud computing, fog computing, dan edge computing.

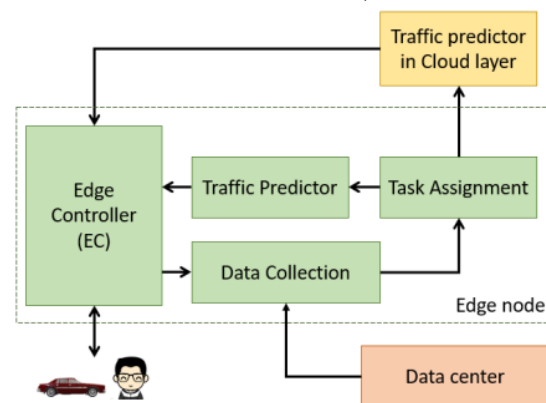


Gambar 1 Level pelayanan setiap jenis server

### III. METODE PENELITIAN

#### A. Pengontrolan Proses Estimasi Waktu Tempuh

Pada penelitian ini, pengendara melakukan request ke edge server atau cloud server untuk diberikan estimasi waktu tempuh ke tujuan. Informasi waktu tempuh dapat diperoleh dari proses estimasi yang dilakukan pada edge server atau cloud server menggunakan informasi yang diperoleh dari komunikasi antar pengendara (Vehicle to Vehicle Communication), pengendara ke infrastruktur (Vehicle to Infrastructure Communication), dan infrastruktur ke infrastruktur (Infrastructure to Infrastructure Communication).



Gambar 3. Blok Diagram Prediksi Trafik

Gambar 3 menunjukkan blok diagram alur data transfer untuk melakukan prediksi keadaan lalu lintas. Sensor IoT yaitu camera dan pengendara mengirimkan informasi ke data center. Informasi kemudian digunakan pada edge server pada bagian data collection untuk menampung permintaan estimasi waktu dari pengendara. Kemudian, permintaan waktu tempuh dan data dari sensor IoT dikirimkan ke task assignment. Task assignment akan memilih permintaan diproses di edge server atau diteruskan ke cloud server. Jika tujuan masih berada pada jangkauan edge server maka, permintaan diproses di edge server. Namun, jika permintaan tidak dalam lingkup edge server yang dimaksud maka permintaan

diproses di cloud server. Selanjutnya, edge server atau cloud server melakukan prediksi trafik dan estimasi waktu tempuh untuk pengendara tersebut.

Pada arsitektur edge server, terdapat beberapa asumsi terkait komunikasi setiap kendaraan dengan kendaraan (vehicle to vehicle communication) dan infrastruktur (vehicle to infrastructure communication) yaitu sebagai berikut.

- Protokol Zigbee, jangkauan transmisi kendaraan ke edge server berjarak 300 meter.
- Camera yang berada pada sisi jalan digunakan untuk memantau situasi lalu lintas.
- Kendaraan mampu mengirimkan data ke server edge menggunakan Global Position System (GPS) atau Diferensial Global Position System (DGPS) untuk mendapatkan posisi geografisnya.
- Melalui komunikasi kendaraan dengan infrastruktur, edge server menyimpan jarak jalan untuk yang normal kondisi (yaitu, kendaraan bergerak dengan kecepatan rata-rata pada jalan, yaitu 60 hingga 75 km/jam dan tidak ada kebisingan atau gangguan).

### B. Prediksi Trafik menggunakan LSTM

Prediksi trafik digunakan untuk mengestimasi waktu tempuh untuk pengendara. Dengan menggunakan LSTM, waktu tempuh dapat dilakukan diperoleh dari edge server dan cloud server. Pertama data yang diperoleh dari data collection berupa informasi kepadatan trafik di titik pengendara saat ini dan titik tujuan pengendara. Jika jumlah pengendara kurang dari 10, maka dinyatakan kondisi sepi atau diberikan bobot 1. Jika jumlah kendaraan antara 10-30, maka dinyatakan kondisi jalan sedang dari kemacetan dan diberikan bobot 2. Kemudian, jika jumlah kendaraan lebih dari 30 maka kondisi jalan dinyatakan macet dan diberikan bobot 3. Selain itu, informasi juga diperoleh dari data center yang berasal dari sensor IoT berupa informasi kondisi jalan atau jumlah kendaraan pada data sebelumnya atau dataset.

Selanjutnya, proses *pre-processing* yaitu menormalisasikan dataset dengan mengkonversi *datetime* untuk memperoleh perubahan *dataset* sebelumnya. Untuk proses *training* dan *testing*, dataset dipisahkan menjadi 75% untuk *training* dan 25% untuk *testing*.

Google cloud platform digunakan untuk membuat pemodelan LSTM menggunakan 4 hidden layer, batch size 32, optimize adam. Dengan menggunakan model ini, proses training dimulai untuk melatih data training. Saat proses training diharapkan weight dan bias yang diperbaharui secara terus menerus mampu menghasilkan model yang sesuai. Model tersebut sebagai hasil dari proses training. Selanjutnya, dengan model pembelajaran yang telah didapat, maka proses testing dilakukan. Hasil dari proses akan menunjukkan akurasi dari prediksi trafik yang telah dipelajari dari model yang dihasilkan.

Pada penelitian ini, terdapat 100 skenario lalu lintas yang digunakan untuk mengevaluasi sistem menggunakan edge server dan cloud server. Kondisi titik jalanan yang padat untuk waktu tertentu seperti 07:00 sampai 09:00 dan 18:00 sampai 20:00 dan kondisi normal ditunjukkan dari pukul 09.00 sampai 12 PM dan 23:00 sampai 07:00. Berdasarkan kondisi tersebut, skenario dirancang secara manual tanpa lampu lalu lintas dan kendaraan tidak berhenti. Pada map dibagi menjadi waktu sibuk dan waktu normal. Di waktu sibuk, map menampilkan banyak kemacetan pada setiap titik atau jalan. Sedangkan waktu normal, map menempatkan banyak jumlah segmen jalan sepi. Kemudian memberikan bobot untuk membedakannya yaitu nilai 1 untuk tidak macet, nilai 2 untuk macet sedang, dan nilai 3 untuk sangat macet. Bobot ini menjadi informasi input yang dibutuhkan oleh edge server atau cloud server untuk mengestimasi waktu tempuh.

Selanjutnya, Estimasi waktu tempuh menggunakan algoritma LSTM dianalisis melalui akurasi metrik yang sesuai dengan sistem yang diusulkan. Untuk perbandingan perkiraan waktu perjalanan, hitung di kami arsitektur dan prediksi lalu lintas menghitung di lapisan awan. Itu akurasi ditunjukkan melalui kesalahan fungsi kerugian, yaitu root mean square error (RMSE), mean absolute error (MAE), dan mean absolute percentage error (MAPE) yang ditulis sebagai berikut.

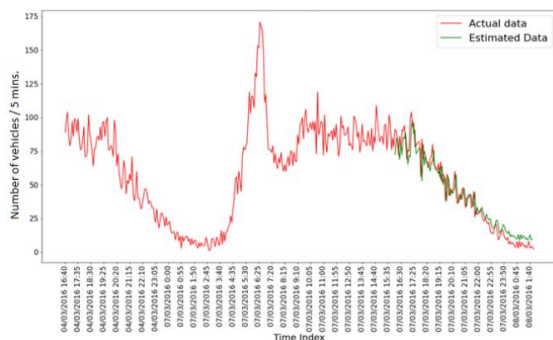
$$MSE = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}, \quad (1)$$

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t|, \quad (2)$$

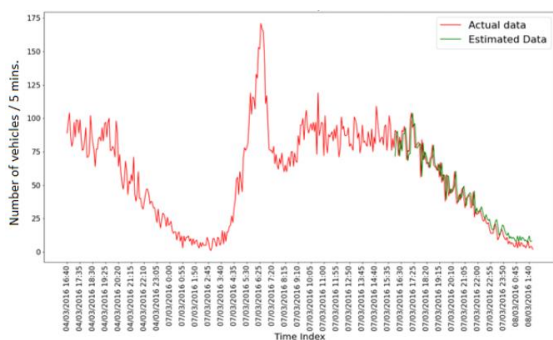
$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right|, \quad (3)$$

**IV. HASIL DAN PEMBAHASAN**

Spesifik hardware yang digunakan pada google cloud platform adalah Python google compute engine backend (GP) dengan RAM 1.63/25.51 GB, disk 31.40/68/40 GB. Dalam percobaan ini, dilakukan dua pengujian LSTM yang berbeda dari sesi dense, yaitu LSTM 1 menunjukkan dense 128 dengan loop back 1 dan LSTM 2 menunjukkan dense 256 dengan loop back 1. Gambar 4 dan 5 menunjukkan performansi prediksi trafik menggunakan LSTM 1 dan 2 secara berurutan.



Gambar 4. Prediksi Trafik menggunakan LSTM 1



Gambar 5. Prediksi Trafik menggunakan LSTM 2

Tabel 2 menunjukkan perbandingan performansi RMSE, MAE, dan MAPE dari LSTM 1 dan LSTM 2. Pada tabel menunjukkan bahwa persentase kesalahan terendah ditunjukkan oleh LSTM 2 yaitu RMSE 10.78%, MAE 8.24%, dan MAPE 19.87%. Hal ini juga menunjukkan bahwa dense yang digunakan juga dapat mempengaruhi akurasi yang terlihat dari persentase kesalahan. Selain itu LSTM 2 menunjukan performansi yang optimal untuk prediksi time secara real-time.

Tabel 1. Perbandingan performansi LSTM 1 & LSTM 2

	RMSE (%)	MAE (%)	MAPE (%)
LSTM1	11.14	8.46	22.11
LSTM2	10.8	8.24	19.87

Perhatian praktis dari penelitian ini adalah wilayah perkotaan yang telah menerapkan konsep IoT pada transportasi, persimpangan, dan jalan kota. Contohnya, Washington D.C., Boston, dan Massachusetts, yang kendaraannya tersebar banyak untuk setiap jalan segmen.

Dengan Metode LSTM, edge server mendukung prediksi trafik lalu lintas dengan akurasi cukup baik karena LSTM berdasarkan deret waktu prediksi dan memori untuk menyimpan data historis. Oleh karena itu, sistem menempatkan kendala pada komputasi yang limit dan penyimpanan yang terbatas pada edge node dan protokol komunikasi untuk kendaraan tertentu (Contohnya, kendaraan yang digunakan pelanggan untuk kerangka ITS edge server yang ditugaskan).

**V. KESIMPULAN**

Penelitian menghasilkan prediksi trafik yang akan digunakan untuk mengestimasi waktu tempuh pada permintaan pengendara yang mengakses edge server. Dengan menggunakan LSTM, prediksi trafik yang dihasilkan cukup baik untuk digunakan, dimana persentase kesalahan yang ditunjukan lebih sedikit untuk LSTM dengan dense 256 dan loopback 1.

**REFERENSI**

- [1] A. Rejeb, K. Rejeb, S. Simske, H. Treiblmaier, and S. Zailani, "The big picture on the internet of things and the smart city: a review of what we know and what we need to know," *Internet of Things*, vol. 19, p. 100565, Aug. 2022, doi: 10.1016/j.iot.2022.100565.
- [2] A. S. Syed, D. Sierra-Sosa, A. Kumar, and A. Elmaghraby, "IoT in Smart Cities: A Survey of Technologies, Practices and Challenges," *Smart Cities*, vol. 4, no. 2, pp. 429–475, Mar. 2021, doi: 10.3390/smartcities4020024.
- [3] L. A. Paramesti and D. Atunggal, "ANALISIS TINGKAT KESESUAIAN DARI KLASIFIKASI

- KEPADATAN LALU LINTAS DAN WAKTU TEMPUH GOOGLE MAPS,” *JGISE: Journal of Geospatial Information Science and Engineering*, vol. 2, no. 2, Dec. 2019, doi: 10.22146/jgise.51134.
- [4] A. Khumaidi, R. Raafiudin, and I. P. Solihin, “Pengujian Algoritma Long Short Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung,” *Jurnal Telematika*, vol. 15, no. 1, pp. 13–18, 2020.
- [5] M. Rizki, S. Basuki, and Y. Azhar, “Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(LSTM) Untuk Prediksi Curah Hujan Kota Malang,” *Jurnal Repositor*, vol. 2, no. 3, pp. 331–338, Mar. 2020, doi: 10.22219/repositor.v2i3.470.
- [6] P. Poonia and V. K. Jain, “Short-Term Traffic Flow Prediction: Using LSTM,” in *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, IEEE, Feb. 2020, pp. 1–4. doi: 10.1109/ICONC345789.2020.9117329.
- [7] I. O. Arissinta, I. D. Sulistiyawati, D. Kurnianto, and I. Kharisudin, “Pemodelan Time Series untuk Peramalan Web Traffic Menggunakan Algoritma Arima, LSTM, dan GRU,” in *PRISMA, Prosiding Seminar Nasional Matematika*, Semarang: Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang, 2022, pp. 683–700.
- [8] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, “iRAF: A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks,” *IEEE Internet Things J*, vol. 6, no. 4, pp. 7011–7024, Aug. 2019, doi: 10.1109/IIOT.2019.2913162.
- [9] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey,” *Future Generation Computer Systems*, vol. 97, pp. 219–235, Aug. 2019, doi: 10.1016/j.future.2019.02.050.