Machine Learning in Sensors for Collision Avoidance

Erkan Karakus, Tao Wei, and Qing Yang
Dept. of Electrical, Computer, and Biomedical Engineering
University of Rhode Island
Kingston, RI, 02881 USA
{erkan_karakus, tao_wei, qyang}@uri.edu

***Abstract*— Sensors generate a huge amount of data that need to be transferred to a computing device for processing. Such large data transfer takes time and consumes energy. This paper presents a new sensing and computing architecture, referred to as MLIS (Machine Learning in Sensors). MLIS allows a part of machine learning to be done on sensor board thereby dramatically reducing the amount of data transferred to the computing device and hence improving overall system performance and energy efficiency. Using an energy-based probabilistic graphical model, RBM (Restricted Boltzmann Machine), we built a new ADAS (Advanced Driver-Assistance System) computing platform for autonomous driving with phased-array-radar as sensors. A working prototype has been built to provide proof of concept for our new architecture. The prototype is implemented using a TI's mmWave (millimeter Wave) radar board and a Vivado HLS implementation of the RBM on the Xilinx xc7z020-clg400-1 device. Extensive experiments have been carried out using the prototype on realistic scenes on our campus. Experimental results have shown that the proposed architecture can reduce the data to be transferred by a factor of 8 while maintaining 98% accuracy. Based on the experimental settings, we present two case studies that have shown a remarkable reduction in collision probability if applying the new architecture to autonomous vehicles.**

Keywords—In-sensor computing, computer architecture, mmWave radar, deep learning, object detection and identification

I. INTRODUCTION

In today's digital world, sensors generate a huge amount of data in a variety of applications. This huge amount of data is generally transferred to the DRAM of computing systems for processing through networks or direct connections. Transferring such a huge amount of data consumes energy and results in long transfer delays. In real-time applications, especially in mission-critical real-time applications such as autonomous vehicles, any delay over the hard deadline implies a life and death situation. Therefore, minimizing data transfer and hence latency is extremely important.

Internal communication and computing in an autonomous vehicle should be designed to provide fault tolerance, energy efficiency, determinism, high bandwidth, and flexibility [1]. End-to-end latency and communication overhead play a critical role to ensure data consistency and temporal determinism across functional cause-effect chains [2], [3].

This paper presents a new sensing, communication, and computing architecture for autonomous vehicles. The new architecture, referred to as MLIS (Machine Learning in Sensors), leverages an energy-based generative model, RBM (Restricted Boltzmann Machine). MLIS involves three major steps: First, sensed raw data go through the first layers of RBM as a "generate phase" on the sensor board. Second, the outputs of the hidden layer units of RBM are transmitted to the central ADAS computer. Third, ADAS computer carries out the phase of 1-step Gibbs sampling as the "reconstruct phase". Because of in-sensor computing of the generate phase, the data transferred from the sensor board to the DRAM of the central ADAS computer is reduced, giving rise to dramatically improved performance and energy efficiency of MLIS.

After establishing the accuracy of MLIS, we built a working prototype using Texas Instruments AWR1243 FMCW mmWave radar board and Xilinx xc7z020-clg400-1 device. Extensive experiments have been carried out using our prototype MLIS on practical scenes on our campus for object detection and decision making process. Experimental results have shown that MLIS yielded a reconstruction accuracy of 98% accuracy, dimensionality reduction of a factor of 8 and reduced the point-to-point data transfer latency by a factor of 6. To demonstrate how such latency reduction helps improve collision avoidance in autonomous vehicle applications, two case studies were performed using our prototype MLIS to show a dramatic reduction in collision probabilities.

This paper makes the following contributions:

1) A new in-sensor computing architecture is proposed on a radar sensor board that implements RBM together with the ADAS computer.
2) A two-phase 1-step Gibbs sampling computation framework consisting of generate phase and reconstruct phase is presented.
3) Extensive experiments have been conducted to demonstrate the feasibility and performance of MLIS. Numerical results have shown up to 8 times reduction in data transfers between the sensor board and central computer with a reconstruction accuracy of 98%.
4) Braking distance gain due to latency reduction is shown to be significant when implementing the generate phase of the MLIS on an FPGA.
5) Our case studies show that the probability of collision is reduced dramatically with the new MLIS architecture.

The paper is organized as follows. Next section presents the basic architecture of MLIS. Section III presents the design of Restricted Boltzmann Machine (RBM) model followed by experimental results in Section IV. We discuss related work in Section V and conclude the paper in Section VI.

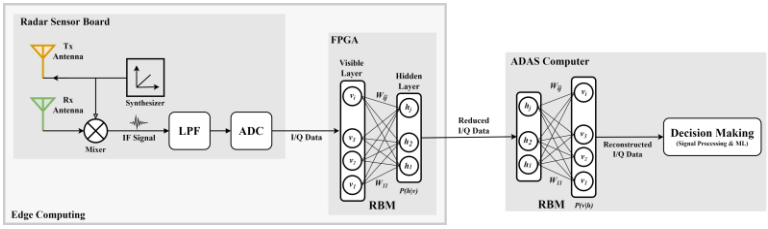


Fig. 1 Proposed Architecture for I/Q Data Reduction and Reconstruction.

II. System Architecture

We propose a new architecture for I/Q data reduction and reconstruction as shown in Fig. 1. The left-hand side of the diagram represents the in-sensor computing platform consisting of the radar sensor and the FPGA board, while the right-hand side of the diagram represents the ADAS computing platform. The implementation of MLIS was split into two distinct phases: Generate and Reconstruct phases as shown in Fig. 2. Later, we will refer to this model as rtl+sw based MLIS since MLIS generate phase is implemented with RTL abstraction model and reconstruct phase is implemented with a software model.

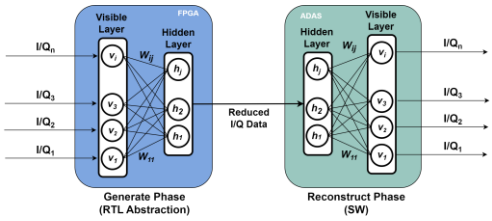


Fig. 2 The Generate and Reconstruct Phases of MLIS.

The generate and reconstruct phase computations are performed using the following equations, respectively.

$$h = v^T W + b \tag{1}$$

$$v = h^T W^T + a \tag{2}$$

where b is the hidden layer units bias vector, a is the visible layer units bias vector, W is the connection weights between the visible and hidden layer units, v and h are visible and hidden layer unit activations, respectively.

III. Results and Discussions

A. *I/Q Data Reconstruction Accuracy*

Fig. 3 and Fig. 4 shows sample plots of true and reconstructed I and Q code data reconstructed by the proposed MLIS architecture, respectively. The original true values are plotted using solid blue lines, the sw based reconstructed data are plotted using dashed red lines and the rtl+sw based reconstructed data are plotted using solid green lines.

To quantify the accuracy assessment, we use Normalized Euclidean Distance (NED) between the two time-series data: reconstructed and true I/Q codes. TABLE I shows the measured statistics for sw and rtl+sw MLIS architectures with different sizes of visible and hidden layer units (v,h). As seen from TABLE I, rtl+sw based MLIS with (512, 64) configuration provides a data reduction factor of 8 with 98.03% accuracy for I code data and 95.7% accuracy for Q code data. For (256, 64) configuration, MLIS provides a data reduction factor of 4 with 98.3% accuracy for I code data and 94.1% accuracy for Q code data.

TABLE I 1-NED Accuracy Statistics for Selected MLIS Architecture Sizes

		Accuracy Statistics (1-NED)%	
(v,h)	**Type**	**I Codes**	**Q Codes**
(512,256)	sw	98.1%	97.2%
	rtl+sw	96.5%	95.3%
(512,128)	sw	98.1%	97.4%
	rtl+sw	97.8%	96.6%
(512,64)	sw	98.4%	96.6%
	rtl+sw	98.0%	95.7%
(512_32)	sw	96.7%	94.4%
	rtl+sw	96.2%	95.0%
(256,128)	sw	98.4%	97.1%
	rtl+sw	98.0%	95.2%
(256,64)	sw	98.4%	95.7%
	rtl+sw	98.3%	94.1%

B. *Data Transfer Latency*

In our experiments, the total amount of data transferred over Ethernet from the data capture board, DCA1000EVM, to the ADAS computer can be computed by multiplying the ADC sample size, loop count, frame count, bit length, LVDS lane count, and channel count. Therefore, transfer latency is given by the following equation:

$$T = \frac{\text{ADC Sample Size} * \text{Loop Count} * \text{Frame Count} * \text{Bit Length} * \text{Lane Count} * \text{Channel Count}}{\text{Bandwidth}} \tag{3}$$

TABLE II shows the comparison in data transfer latencies for different ADC sample size configurations. The first two columns correspond to the original configuration with ADC sample sizes of 512 and 256, respectively. The last three columns correspond to the case where MLIS is deployed to reduce the dimensionality of ADC sample size to 128, 64, and 32, respectively.

TABLE II Data Transfer Latencies For Different ADC Sample Size Cases

	Original		**MLIS**		
ADC Sample Size	512	256	128	64	32
Loop Count	128				
Frame Count	10				
Bit Length	16		22		
Lane Count	4				
Channel Count (I/Q)	2				
BW (Mbps)	600				
Tr. Latency (ms)	140	70	48	24	12

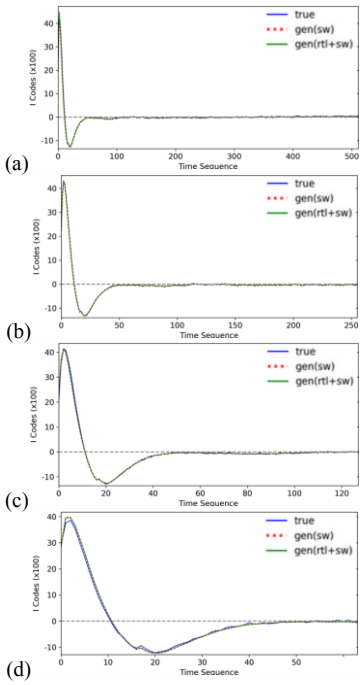


Fig. 3 True and reconstructed (sw and rtl+sw) I codes with ADC sample sizes of (a)512 (b)256 (c)128 (d)64

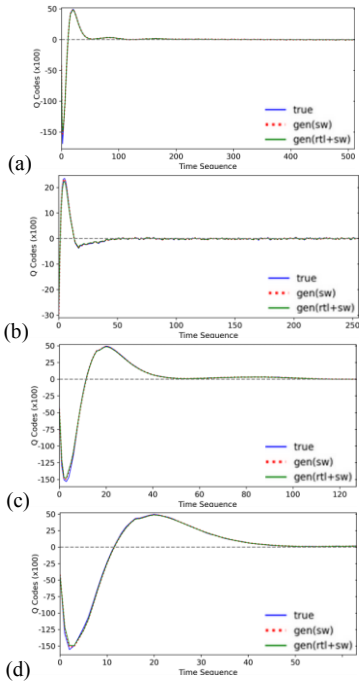


Fig. 4 True and reconstructed (sw and rtl+sw) Q codes with ADC sample sizes of (a)512 (b)256 (c)128 (d)64

The bit length used for I/Q samples is 16-bit in the original case and 22-bit in the MLIS case with the FPGA implementation since we use ap_fixed<22,16> data format on RTL design. The average data transfer rate between the data capture board, DCA1000EVM, and ADAS computer is 600 Mbps. Each ADC sample is represented by a complex data format, consisting of a real (I code) and imaginary part (Q code) and each LVDS lane captures the complex data samples per receiver antenna. The AWS1243 radar board has 4 LVDS lanes and each lane receives I and Q codes, thus the channel count is 2. As shown in TABLE II, the achieved latency reduction (speedup) ranges from 46% (70/48) up to an order of magnitude (140/12).

C. *The Model Processing Latencies*

The total end-to-end data reduction and reconstruction latency is the sum of data transfer latency and model processing latency. TABLE III shows the model processing latencies of the architecture on the FPGA board (Vivado HLS FPGA) and ADAS CPU, respectively. The model processing latencies on the FPGA board (Vivado HLS FPGA) takes in the range of nanoseconds while the processing time on the ADAS CPU is on average 17 milliseconds as shown in TABLE III. TABLE IV shows the reduction in total latency including the latency of transferring the reduced radar data from the FPGA board to the ADAS computer and the latency of model processing.

D. *Probabilistic Analysis of Collisions*

The Collision Avoidance System (CAS) and Automated Emergency Braking System (AEBS) are the crucial functions of ADAS applications. The probability of non-collision with an object can be described as the probability of the braking distance being less than the object distance. We assume that the braking distance has a variance due to different factors such as tire tread, braking mechanics condition etc. The probability of collision can be expressed as

$$P(Collision) = 1 - P(d_b < d_o \mid \mathcal{V} = v) \quad (4)$$

where d_b and d_o are braking and object distance, respectively. v is the vehicle speed. The mean value of braking distance is given by the following equation [4].

$$\mu_{bd} = \frac{v^2}{2\mu g} \quad (5)$$

where v (m/s) is the velocity, μ is friction coefficient, g (m/s^2) is the acceleration due to gravity. Then the deceleration is given by

$$a = -\mu g. \quad (6)$$

TABLE III. RTL AND SW MODEL EXECUTION LATENCIES ON HLS AND ADAS CPU

ADC Sample Size	Reduced Size	*rtl* model absolute processing latency(ns) Vivado HLS	*sw* model average processing latency(ms) ADAS CPU
256	32	28.567	17
256	16	1.430	17
128	32	0.87	17
128	16	0.78	17

TABLE IV. DATA TRANSFER AND PROCESSING LATENCY REDUCTION

	Original/Reduced Data Sample Size			
	256/32	256/16	128/32	128/16
Data Transfer Latency Reduction (ms)	58	64	23	29
Processing Latency Reduction (ms)	17	17	17	17
Total Latency Reduction (ms)	75	81	40	46

E. *Case Studies for Collision Avoidance*

We present two different collision scenarios where AEBS is engaged immediately to prevent any collision. We use the deceleration profiles given in TABLE V for the two scenarios. Braking distances were computed by using the formula given in [4].

TABLE V DECELERATION PROFILES FOR TWO CASES

Case	I	II
Deceleration (m/s²)	-8.83	
Starting Velocity (kph)	50	70
Final Velocity (kph)	0	10
Condition	Dry roadway	Dry roadway
Braking Distance (m)	11	21

Case I: Collision Avoidance at Pedestrian Crossing:

Let us consider the scenario depicted in Fig. 5 in which a car is cruising at a certain speed, while a pedestrian emerges suddenly from the front of a minivan to cross the street. Since the pedestrian is occluded by the minivan, the driver of the car is not capable of realizing the presence of the pedestrian in front of the minivan. This may lead to a serious collision with the pedestrian that requires AEBS to be engaged immediately to prevent an imminent collision with the pedestrian. If the car is cruising at a speed of 50 kph and if the deployed MLIS is 512/64, then the distance gained is 1.8 meters. As soon as the car detects the pedestrian on its cruise way, the ADAS will engage the AEBS to stop the car.

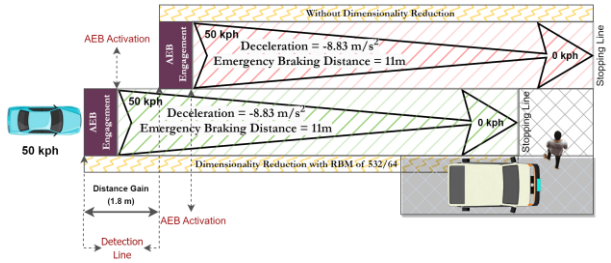
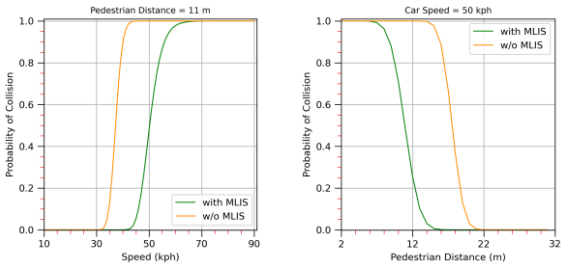


Fig. 5 Case I: AEBS with Pedestrian Crossing Scenario

After AEBS is engaged, the car will decelerate and stop after 11 m preventing any collision. The car would have traveled 1.8 meters farther wwithout MLIS, causing a collision.



(a) (b)

Fig. 6 Case I: Collision Probability with and w/o MLIS (a) for different car speeds. (b) for different pedestrian distances while cruising at 50 kph

Using our probability model above and assuming a standard deviation of 15% of its mean, Fig. 6 (a) shows the probability of collision with and without MLIS for different speeds when the pedestrian is 12 meters away from the car when detected by the radar sensor board. As seen from Fig. 6 (a), the collision probability is lower with MLIS than that without MLIS for the same car speed. Fig. 6 (b) shows the probability of collision with and without MLIS for different pedestrian distances while the vehicle cruises at 50 kph.

Case II: Collision Avoidance with a Leading Car:

In this scenario, a car is cruising at the speed of 70 kph behind an SUV. The driver of the SUV makes a sudden brake to avoid a collision with an object in front of it. The car behind the SUV detects its rapid deceleration and engages the FCW system to alert the driver of the hazard. After the FCW engagement, the driver of the car starts braking until the car is distanced securely from the decelerating SUV. This scenario is depicted in Fig. 7. The MLIS with 512/64 configuration helps gain 2.6 meters of distance, greatly reduced the chance of collision. Fig. 8 (a) shows the probability of collision with and without MLIS for different speeds when the leading SUV is distanced at 21 meters away from the car when detected by the radar sensor board. As seen in Fig. 8 (a), the probability of collision is substantially lower with MLIS than that without MLIS for the same car speed. As the speed increases the probability of collision increases as expected. Similar observations are shown in Fig. 8 (b) for different leading SUV distances while the vehicle cruises at 70 kph.

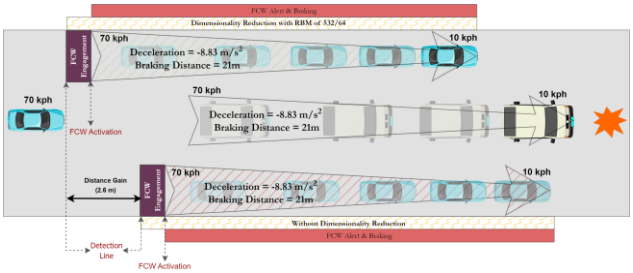
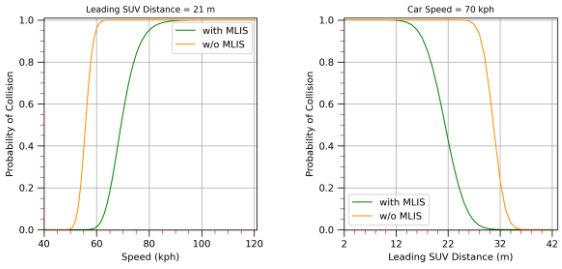


Fig. 7 Case II: Forward Collision Warning Scenario



(a) (b)

Fig. 8 Case II: Collision Probability with and w/o MLIS (a) for different car speeds. (b for different leading SUV distances while cruising at 70 kph.

IV. RELATED WORK

The major challenge with edge computing is the limited resources available on the edge devices and a large amount of sensed raw data to be transferred to the main computer for further processing [5], [6]. Any latency may become a major bottleneck in real-time applications which have hard real-time computation requirements [7]. High Correlation Filter,

Principal Component Analysis, General Discriminant Analysis etc. are among the techniques used for the dimensionality reduction [8]. Vanilla Autoencoders and Convolutional Autoencoders are commonly used in deep neural networks to remove noise and redundant information in high-dimensional data [9]. RBM has captured researchers' interest and many researchers produced hardware designs of the RBM model on FPGAs [10], [11]. The field of time series forecasting has received significant interest in academia and has a wide range of applications in the areas of energy, communication, business, finance, health, and sports [12], [13], [14], [15], [16], [17].

Our work in this paper differs from the studies discussed above in many substantial ways. First, our work concentrates on having a portion of machine learning computations on the radar sensor board to minimize necessary data to be transferred from the radar sensor board to the computing device. Secondly, to reconstruct the approximated data by applying 1-step Gibbs sampling, we implemented the RBM model on the FPGA and its reverse symmetric model on computing device. Third, we carried out extensive experiments to demonstrate the advantages of MLIS in a real-time application: ADAS system in autonomous vehicles. Furthermore, a probabilistic collision avoidance model showed that the probability of collision decreases dramatically with our MLIS architecture.

V. Conclusions

We introduced a novel in-sensor machine learning system, MLIS, applicable to ADAS in autonomous vehicles. The idea is to reduce the amount of data transferred from the sensor board to the main computing platform by performing machine learning computations on the sensor board. The new architecture has shown itself to be capable of substantially reducing the time for decision making, which is critical for real time applications. We used the Texas Instruments' AWR1243 FMCW radar board and Vivado HLS to implement an experimental prototype. Experimental findings showed that it is possible to reduce data transferred from the sensor board to the central computer by a factor of up to 8 with an accuracy of 98%. The collision probabilistic model showed that the data reduction dramatically reduced the collision probability for two collision cases because of the gain in reaction time of ADAS thus increasing the vehicle agility to prevent serious collisions in the most severe cases.

Acknowledgment

This research is supported in part by the National Science Foundation (NSF) under grants 2027069 and 2106750. The authors thank the anonymous reviewers for their comments and suggestions.

References

[1] T. Nolte, H. Hansson, and L. Lo Bello, "Automotive communications - Past, current and future," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2005, pp. 985–992. doi: 10.1109/etfa.2005.1612631.

[2] A. Hamann, D. Dasari, S. Kramer, M. Pressler, and F. Wurst, "Communication centric design in complex automotive embedded systems," in *29th Euromicro Conference on Real-Time Systems (ECRTS 2017)*, 2017, pp. 10:1-10:20. doi: 10.4230/LIPIcs.ECRTS.2017.10.

[3] M. Becker, D. Dasari, S. Mubeen, M. Behnam, and T. Nolte, "End-to-end timing analysis of cause-effect chains in automotive embedded systems," *Journal of Systems Architecture*, vol. 80, pp. 104–113, Oct. 2017, doi: 10.1016/j.sysarc.2017.09.004.

[4] M. Sabri and A. Fauza, "Analysis of vehicle braking behaviour and distance stopping," *IOP Conf Ser Mater Sci Eng*, vol. 309, no. 1, p. 012020, Feb. 2018, doi: 10.1088/1757-899X/309/1/012020.

[5] E. Oyekanlu, "Predictive edge computing for time series of industrial IoT and large scale critical infrastructure based on open-source software analytic of big data," in *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 1663–1669. doi: 10.1109/BigData.2017.8258103.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things J*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[7] D. Fernandez, C. Gonzalez, D. Mozos, and S. Lopez, "FPGA implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images," *J Real Time Image Process*, vol. 16, no. 5, pp. 1395–1406, Oct. 2019, doi: 10.1007/s11554-016-0650-7.

[8] S. Velliangiri, S. Alagumuthukrishnan, and S. I. Thankumar Joseph, "A Review of Dimensionality Reduction Techniques for Efficient Computation," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 104–111. doi: 10.1016/j.procs.2020.01.079.

[9] E. Karakus, M. Bruckner, T. Wei, and Q. Yang, "In-Sensor Neural Network Preprocessing for ADAS Computer Systems," in *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*, IEEE, Oct. 2022, pp. 1–8. doi: 10.1109/NAS55553.2022.9925497.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science (1979)*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[11] D. L. Ly and P. Chow, "A high-performance FPGA architecture for restricted boltzmann machines," in *Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays - FPGA '09*, 2009, pp. 73–82. doi: 10.1145/1508128.1508140.

[12] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra, "Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study," *Chaos Solitons Fractals*, vol. 140, p. 110227, 2020.

[13] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing Journal*, vol. 90, p. 106181, 2020, doi: 10.1016/j.asoc.2020.106181.

[14] M. Mishra, J. Nayak, B. Naik, and A. Abraham, "Deep learning in electrical utility industry: A comprehensive review of a decade of research," *Eng Appl Artif Intell*, vol. 96, p. 104000, 2020, doi: 10.1016/j.engappai.2020.104000.

[15] A. Baarnhielm, "Multiple time-series forecasting on mobile network data using an RNN-RBM model," Uppsala University, 2017. [Online]. Available: http://www.teknat.uu.se/student

[16] B. Denton, Frank and Feaver, Christine and Spencer, "Time series analysis and stochastic forecasting: An econometric study of mortality and life expectancy," *J Popul Econ*, vol. 18, no. 2, pp. 203–227, 2005.

[17] Y. Hu, J. N. K. Liu, J. You, and P. W. Chan, "Continuous RBM based deep neural network for wind speed forecasting in Hong Kong," *Proceedings of the 2015 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2015*, pp. 368–374, 2015.