AIAA AVIATION Forum
June 27-July 1, 2022, Chicago, IL & Virtual
AIAA AVIATION 2022 Forum

10.2514/6.2022-3605

Wind Field Prediction in Urban Spaces for small Unmanned Aerial Systems using Convolutional AutoEncoders

Rohit K.S.S. Vuppala * and Kursat Kara†
School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078, USA

We attempt to generate Reduced Order Models to predict wind data for testing of small Unmanned Air Vehicles (UAV) or Unmanned Aircraft Systems (UAS) in the context of the urban environment. We aim to utilize and explore two different methodologies using specialized techniques and networks in machine learning like Convolutional Auto-Encoders and Long Short-Term Memory networks to create a framework for prediction of wind field data. We intend to use the data generated from Large Eddy Simulations as high-fidelity input data for training the above mentioned ML techniques and evaluate their performance in generating wind data in a simple urban environment setup. The objective of this work is to compare the two methods and ultimately develop a robust framework for close to real-time or real-time prediction of wind field in urban spaces.

I. Introduction

In the recent years there has been an unprecedented level of growth in use of Unmanned Aerial Vehicles (UAV) for numerous diverse applications. They are being used for mission profiles ranging from disaster management[1], remote sensing[2], agriculture[3], law enforcement[4], 3D mapping of complex terrains[5], urban mapping[6] to in near future for fast moving consumer goods delivery and food catering services[7]. Isolated operation of Unmanned Aerial Vehicle is now outdated and is slowly being replaced with an Unmanned Aerial Systems (UAS) for enhancing their ease of operation and increased efficiency. Although increasing complexity, these systems incorporate enhancements for UAV operation like using local weather data, topology information, path planners and utilizing ground control stations (GCS), command/communication links for their remote or autonomous operation.

With extensive penetration of Unmanned Aerial Vehicle (UAV) operations in civilian applications, there has been a growing need for UAS implementation in urban areas. In order to extensively use urban air spaces, there has been significant research into enabling high-density UAS operations[8][9]. This imparts significant size and weight restrictions leading to small Unmanned Aerial Systems (sUAS). However, due to their minimal size and weight they are highly susceptible to both local turbulent effects like building wakes and meso-scale atmospheric effects in urban boundary layers. Notable research has been done in the recent years into urban effects in applications like climate predictions[10][11], and recently for planning and mitigation of urban heat islands[12][13] but only limited amount into investigation of their influence on flight planning, control and trajectories [14][15][16]. There is significant need for urban wind-field data to develop various control strategies to minimize the effects. While, Reynolds-Averaged Navier-Stokes(RANS) equations could be used to simulate the urban flow fields, they do not depict the unsteady nature of the flow, especially in constricted spaces like urban canyons and dense urban skylines. Large Eddy Simulations (LES), unlike the RANS provide a more accurate depiction, closely resembling realistic wind conditions in atmopsheric and urban boundary layers[17][18][19]. But, Large Eddy Simulations are computationally more expensive than RANS and any real-time wind aware strategies cannot be widely adopted.

Reduced Order Models (ROMs) could be key enablers and important part of developing wind-aware strategies and necessary framework for UAS operation. They have been used in diverse applications ranging from process simulation and optimisation [20], flow control [21] to fluid flows[22]. With advent of machine learning, Machine Learning based Reduced Order Models (ML-ROMs) could be generated with relative ease and Non-Intrusively[23]. In our previous work we used Proper Orthogonal Based Decomposition (POD) coupled with LSTM networks to generate Non-Intrusive Reduced Order Models (NIROM)[24][25]. Convolutional Auto Encoders (CAE) could be used to project the data to a latent space similar to POD, make predictions and project it back to the spatio-temporal physical space [26][27][28].

*Graduate Research Assistant, rvuppal@okstate.edu; Student Member AIAA
†Assistant Professor, kursat.kara@okstate.edu; Senior Member AIAA.

Convolutional Neural Networks allow the spatial information to be learned by the neural networks by applied self learned convolutional filters on the training data. This spatial information could deem to be helpful to make accurate predictions. The modes from latent space could be trained on specialized networks like LSTMs to make predictions in time[29] or the CAE could directly be used to make predictions. We explore the two methods and make observations for a simple case of flow across a building.

II. Methodology

In this section we discuss the methodology for our approach which includes generating Large Eddy Simulation data from numerical solvers and, the Non-intrusive Convolutional Auto-Encoders/Convolutional Auto-Encoders coupled with LSTM approaches to predict the flow field in a given domain of interest.

A. LES simulation setup

Large Eddy Simulation data is obtained using Parallelized Large-Eddy Simulation Model (PALM)[30]. PALM is a turbulence-resolving, Large Eddy Simulation solver for atmospheric and oceanic boundary-layer flows. The model is based on solving non-hydrostatic, filtered, incompressible Navier-Stokes equations in Boussinesq-approximated form on a cartesian grid. Implicit separation of sub-grid scales and resolved scales is achieved by averaging the governing equations over discrete grid volumes as proposed by Schumann[31].

1. Governing Equations

The model solves for six prognostic quantities, the velocity components u, v, w, the potential temperature θ, specific humidity q_v and the SGS turbulent kinetic energy e. The potential temperature is defined as

$$\Theta = \frac{T}{\Pi} \tag{1}$$

from absolute temperature T and the Exner function,

$$\Pi = \left(\frac{p}{p_0}\right)^{\frac{R_d}{C_P}} \tag{2}$$

where, p is the hydrostatic pressure, p_0 is the reference pressure 1000 hPa, R_d is the gas constant for dry air and C_P is the specific heat of dry air at constant pressure. Furthermore a virtual potential temperature could be calculated using the relation,

$$\Theta_v = \Theta\left[1 + \left(\frac{R_v}{R_d} - 1\right) q_v - q_l\right] \tag{3}$$

where R_v is the gas constant for water vapor and q_l is the liquid water specific humidity calculated based on a chosen cloud micro-physics model. (Note: for the present study dry atmospheric boundary conditions with neutral stratification are considered eliminating the need for cloud multi-physics and also making the absolute temperature, potential temperature and virtual potential temperature the same value) The governing equations for the conservation of mass, momentum, energy and moisture filtered over a cartesian grid are expressed below in Einstein summation notation, where angle brackets denote horizontal domain average, over-bar indicates filtered quantities and double-prime indicates SGS variables.

$$\frac{\partial \overline{u_i}}{\partial t} = -\frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} - \varepsilon_{ijk} f_j \overline{u_k} + \varepsilon_{i3j} f_3 \overline{u}_{g,j} - \frac{1}{\rho_0}\frac{\partial \Pi^*}{\partial x_i} + g\frac{\overline{\Theta_v} - <\overline{\Theta_v}>}{<\overline{\Theta_v}>}\delta_{i3} - \frac{\partial\left(\overline{u_i'' u_j''} - \frac{2}{3}\overline{e}\delta_{ij}\right)}{\partial x_j} \tag{4}$$

$$\frac{\partial \overline{u_j}}{\partial x_j} = 0 \tag{5}$$

$$\frac{\partial \overline{\Theta}}{\partial t} = -\frac{\partial \overline{u_j}\overline{\Theta}}{\partial x_j} - \frac{\partial\left(\overline{u_j'' \Theta''}\right)}{\partial x_j} - \frac{L_v}{C_p \overline{\Pi}}\Psi_{q_v} \tag{6}$$

$$\frac{\partial \overline{q_v}}{\partial t} = -\frac{\partial \overline{u_j q_v}}{\partial x_j} - \frac{\partial \overline{u_j'' q_v''}}{\partial x_j} + \Psi_{q_v} \tag{7}$$

where,
$u_i (i = 1, 2, 3)$ represents the components of velocities, f_i is the Coriolis parameter, L_v is latent heat of vaporisation, g is the gravitational acceleration, $u_{g,k}$ are the geostrophic wind components, ρ_0 is the density of dry air, p^* is the perturbation pressure, $\Pi^* = p^* + 2/3\rho_0 e$ is the modified perturbation pressure, and SGS TKE is represented by e.

2. Turbulence closure

The closure includes a prognostic equation for the filtered SGS-TKE $\overline{e}$ given below, the SGS terms are parametrized using 1.5 order closure following Deardorff[32] , using a modified version of Wyngaard et al.[33] and Saiki et al.[34]. For further information regarding the parameterization of various terms in the equation, the reader is referred to [30].

$$\frac{\partial \overline{e}}{\partial t} = -\overline{u_j}\frac{\partial \overline{e}}{\partial x_j} - \overline{u_i'' u_j''}\frac{\partial \overline{u_i}}{\partial x_j} + \frac{g}{\overline{\Theta_{v,0}}}\overline{u_3'' \Theta_v''} - \frac{\partial \left[\overline{u_j'' \left(e + \frac{p''}{\rho_0}\right)}\right]}{\partial x_j} - \epsilon \tag{8}$$

where,
ϵ is the SGS dissipation rate.

B. Non-Intrusive Convolutional Auto-Encoder Methodology

1. CAE-LSTM approach

As outlined in Algorithm 1, we initially obtain the time dependent modal coefficients by performing a Convolutional Encoder transformation on the snapshot data of the velocity fluctuation field obtained from the LES data. An optimal number of these modal coefficients are chosen in the latent space encoded by the trained Convolutional Auto Encoder network. We could now learn these modes using Neural networks to make predictions in time. Recurrent Neural Networks are a widely used neural network architecture in cases where the output information is dependent on current input as well as characteristics learnt from previous observations. RNNs contain cyclic or recurrent connections that enable them to continuously learn characteristics from a series of data and predict future outcomes. We use Long Short-Term Memory (LSTM) neural networks[35], a special variant of RNN architecture better suited for learning long-term dependencies in the input data. After the network is trained we predict the modal coefficients for required number of snapshots. These are then used as input to the decoder to project the modes back to fluctuation field snapshots. We can then calculate the data field by summing it to the previous average calculated.

2. CAE only approach

Unlike previous case where the Encoder and Decoder are trained on the same snapshot as both the input and output data. We instead employ the CAE to make prediction for the next timestep based on current snapshot information. We closely follow the methodology as described in Algorithm 2 to generate the "CAE only" based ML-ROM. This method is relatively easier to train compared to CAE-LSTM network since it has lesser number of parameters to train. Similar Encoder, Decoder structures were used for both the cases to make better comparison for their predicted data.

Algorithm 1 CAE-LSTM approach

1: Obtain solution data from Large Eddy Simulations for the domain of interest.
2: Compute the fluctuation flow field for the given number of snapshots at the 2D region of interest, i.e. mean-subtracted flow field, at center of domain and xz plane

$$\overline{u}(x, y, z, t_n) = \frac{1}{N}\sum_{n=1}^{N} u(x, y, z, t_n)$$
$$u'(x, y, z, t_n) = u(x, y, z, t_n) - \overline{u}(x, y, z, t_n)$$

3: Train the Convolutional Auto-Encoder network with the snapshots of the data from the fluctuation flow field, scaled appropriately for optimal performance.
4: Compute the modal coefficients for the data matrix over time from the encoder portion of the Auto-Encoder.
5: Pre-process the data by scaling and re-arranging data for LSTM training with appropriate look-back window.
6: Predict the modal coefficients with the trained network for future snapshots.
7: Using the decoder portion of the Auto-Encoder, project the modal coefficients back to the spatio-temporal domain.
8: Compute the predicted flow field by adding the mean value to the predicted snapshot data.

Algorithm 2 CAE Only approach

1: Obtain solution data from Large Eddy Simulations for the domain of interest.
2: Compute the fluctuation flow field for the given number of snapshots at the 2D region of interest, i.e. mean-subtracted flow field, at center of domain and xz plane

$$\overline{u}(x, y, z, t_n) = \frac{1}{N}\sum_{n=1}^{N} u(x, y, z, t_n)$$
$$u'(x, y, z, t_n) = u(x, y, z, t_n) - \overline{u}(x, y, z, t_n)$$

3: Train the Convolutional Auto-Encoder network with the snapshots of the data from the fluctuation flow field, scaled appropriately for optimal performance. However, the expected output is the next snapshot of data in time, rather than current snapshot like the CAE-LSTM approach.
4: Compute the predicted flow field by adding the mean value to the predicted snapshot data.

III. Results and Discussion

In this section we briefly talk about the simulation setup for generating the Large Eddy Simulation data for generating the ML-ROMs and additionally compare the results between the two different methodologies adopted for generating the ML-ROM.

A. Simulation Setup

A cubic building of height H in a three-dimensional computational domain is used for generating the Large Eddy Simulation data. For setting up the CFD domain to prevent the influence of boundary conditions, we closely follow the recommendations of Franke et al.[36], Murakami and Mochida[37]. Furthermore the flow conditions at the inlet follow the recommendation of [37] where a vertical profile proportional to $z^{1/4}$ till a height of 2H is used, similar to the setup of Tutar and Oguz was used for x-component of velocity u. The other two components v, w are set to zeros, so the flow is dominant in x-direction. Isotropic mesh of resolution $H/10$ was used. The wind velocity at height H is set to be as $8m/s$, similar to typical high wind conditions. More details about the domains used for CFD is tabulated in table 1 and also depicted in Fig 1. Neutral and dry atmospheric conditions were chosen with Coriolis parameter of $7.3 \times 10^{-}5$ with boundary conditions on the top and bottom (z-direction) as free-slip and no-slip, left and right (x-direction) as inflow and outflow, front and back (y-direction) as outflows respectively.

Domain size	Specification
upstream (x-direction)	2H
downstream (x-direction)	7H
lateral (y-direction)	2.5H
above building (z-direction)	5H

Table 1 Domain details

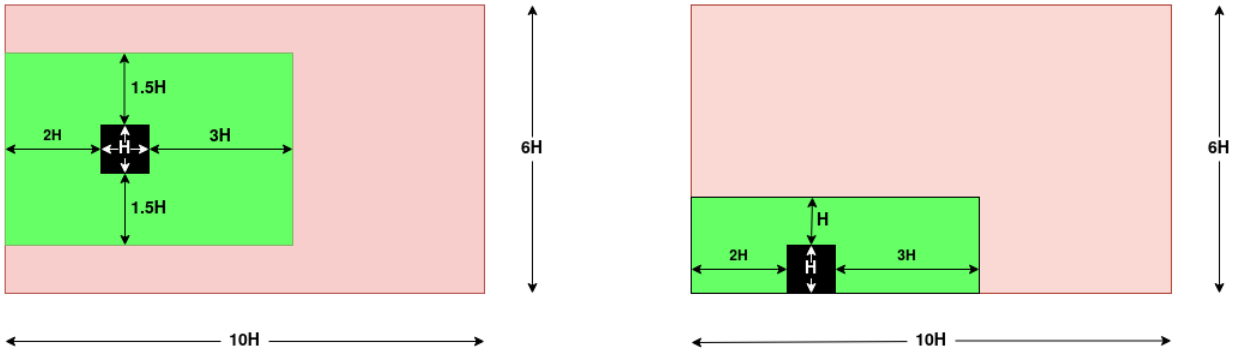


Fig. 1 Left-Top view (xy-plane); Right-Side view(xz-plane) of the domain;
Green-domain of interest where 2D snapshots(xz plane) at center of domain are taken , Red-total domain for LES

The data used for ML-ROM was only from the xz plane in the center of the domain and for simplicity only the u-velocity component is chosen to demonstrate the method. Two different approaches are tried below, one using CAE-LSTM and one using only CAE and both the results are presented. The models are trained for 600 snapshots of data and predictions are made for 600 future snapshots. Similar Encoder and Decoder structures were used across the both the methods as shown in table 2. For both the methods terrain map of 2D domain is given in a secondary channel to the CAE. The information on the building's location in the 2D snapshot is provided as 0's on grid points in absence of the building and 1's for grid points inside building.

Parameter	Specifications
Encoder	[4,(8,8,8),(16,16,16),16,16,8,4] (2 Channel)
Decoder	[8,16,20,24,32]
Modes	24
Skip connections	2[Encoder only, segments depicted by "()"]
Skip connection 1	after 1 to before 5
Skip connection 2	after 1 to before 8

Table 2 CAE details

B. CAE-LSTM Results

Parameter	Specification
Number of hidden layers	2
Number of neurons in each hidden layer	64
Activation function	tanh
Lookback time-window	20
Recurrent dropout	0.2
Neuron dropout	0.2
Loss function	MSE
Optimiser	ADAM
Training-testing ratio	4:1

Table 3 LSTM Neural Network details

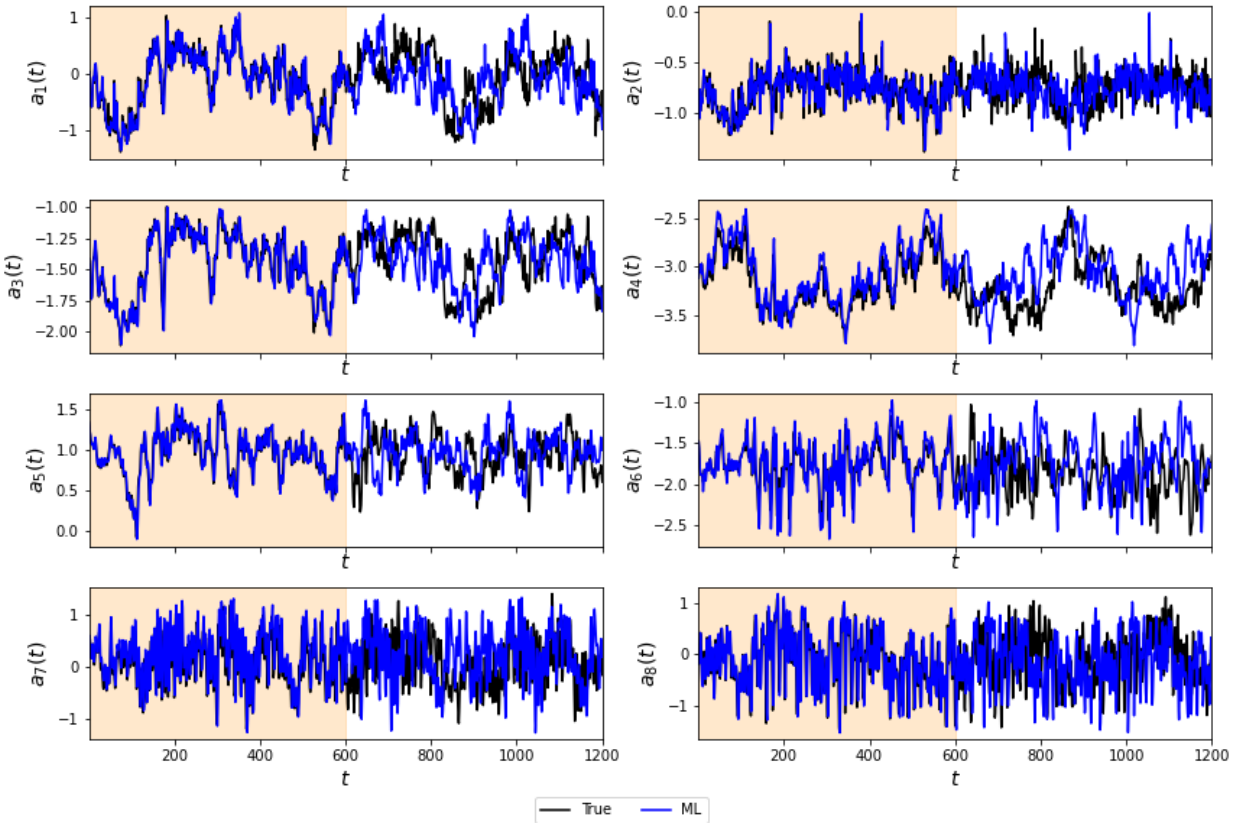
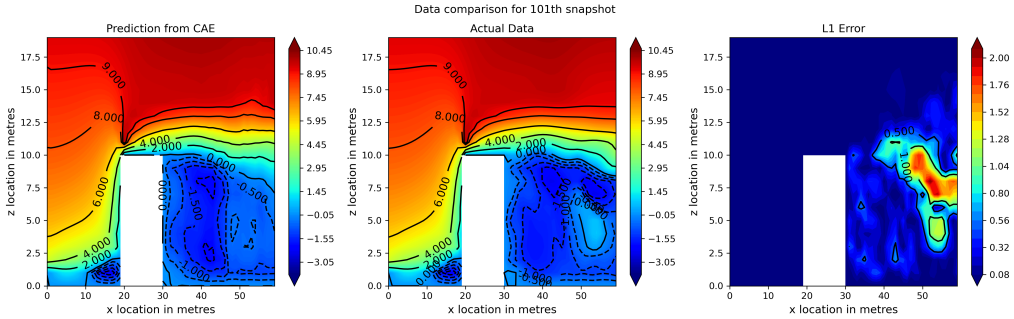
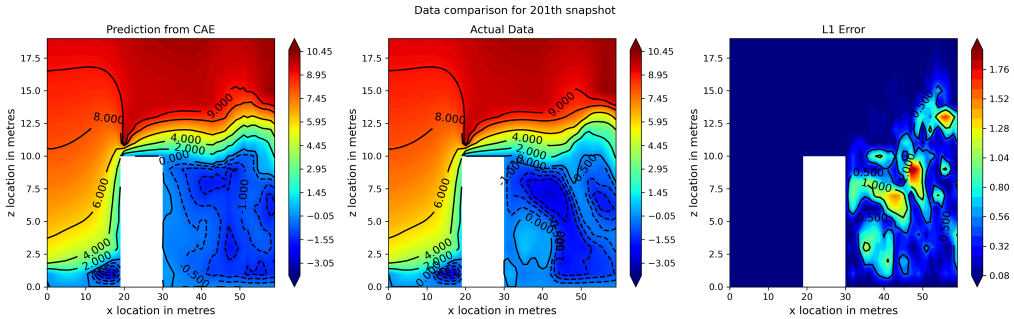


Fig. 2 Comparison between True and CAE-LSTM (ML) for first 8 modes for demonstration; Background colors: Tan/Orange - Training, White - Prediction

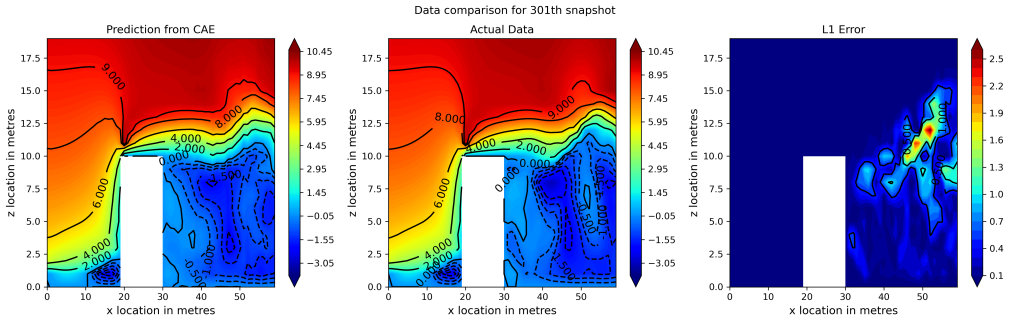
We notice good agreement between the results and the actual data in the predictions from the auto encoder coupled with LSTM. We also notice the predictions for the 24 modes in the latent space is also in good agreement as shown in figure 2. We do notice minor differences in the velocity field predicted. This is expected and could be attributed to the errors from both the LSTM network and the Auto-Encoder.



(a) u-velocity contour for xz plane in center of domain for 100th snapshot

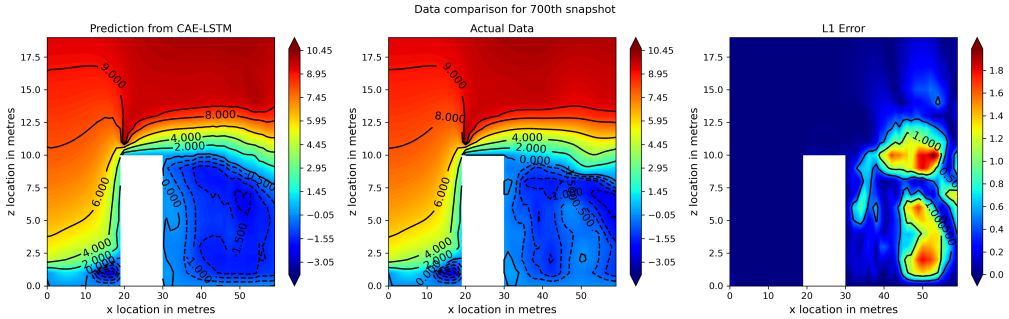


(b) u-velocity contour for xz plane in center of domain for 200th snapshot

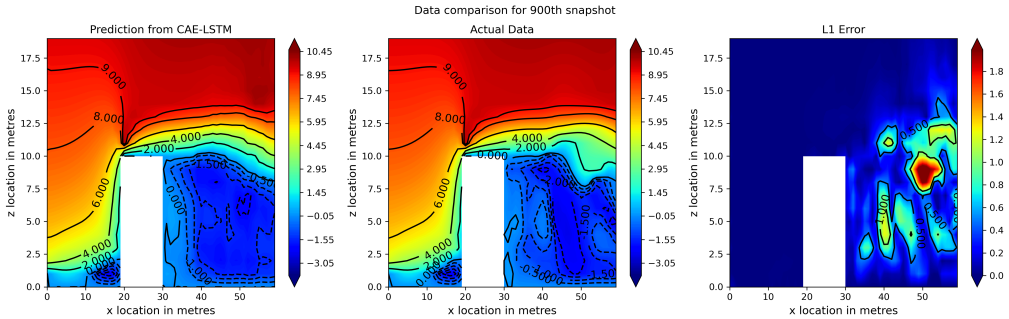


(c) u-velocity contour for xz plane in center of domain for 300th snapshot

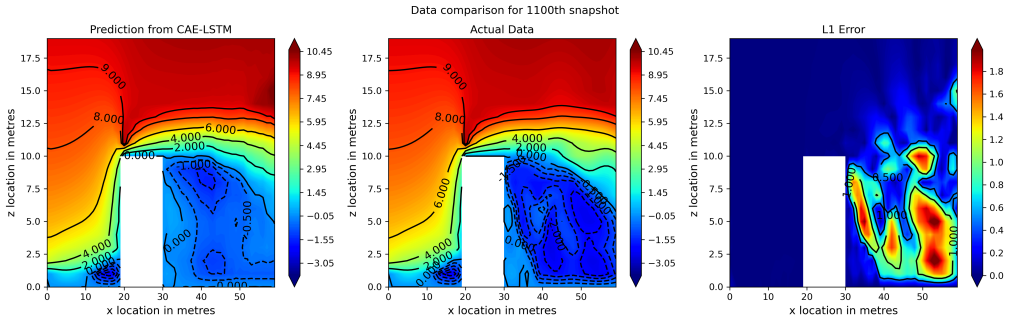
Fig. 3 u-velocity contour for xz plane in center of domain for training phase of AutoEncoder



(a) u-velocity contour for xz plane in center of domain for 700th snapshot



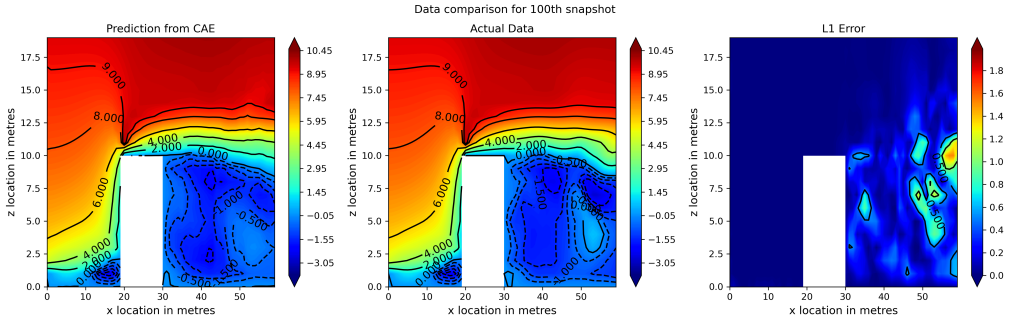
(b) u-velocity contour for xz plane in center of domain for 900th snapshot



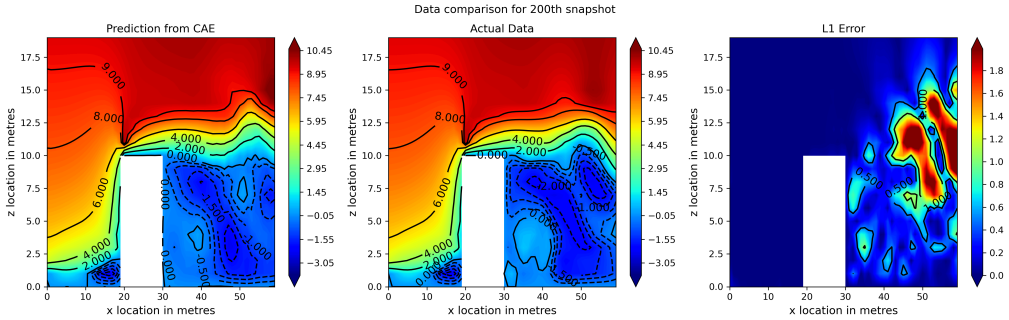
(c) u-velocity contour for xz plane in center of domain for 1100th snapshot

Fig. 4 u-velocity contour for xz plane in center of domain for prediction phase

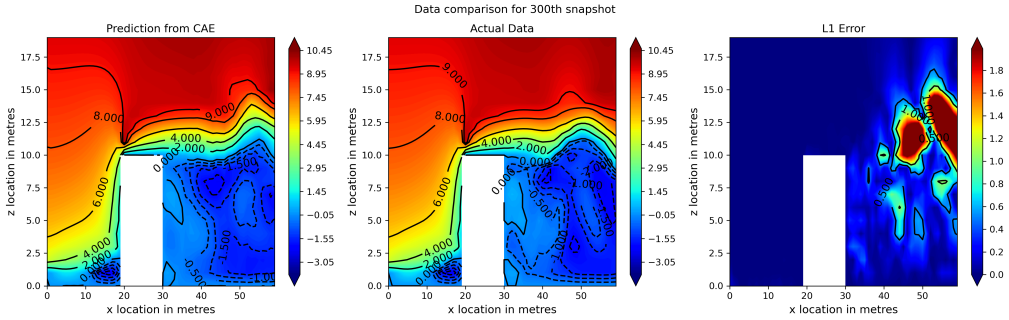
C. CAE Only results



(a) u-velocity contour for xz plane in center of domain for 100th snapshot



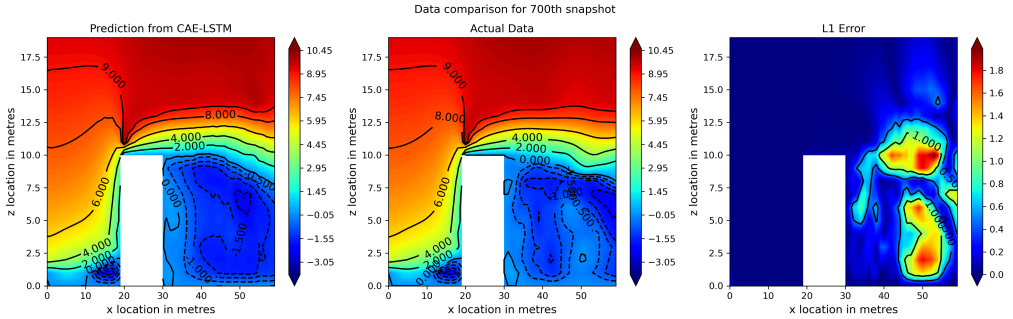
(b) u-velocity contour for xz plane in center of domain for 200th snapshot



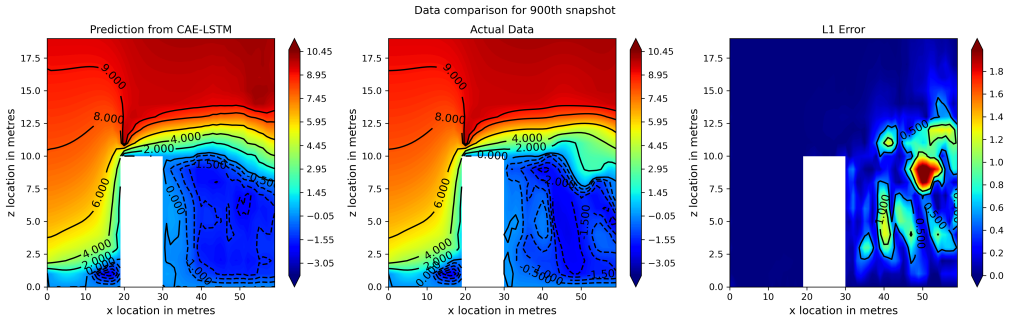
(c) u-velocity contour for xz plane in center of domain for 300th snapshot

Fig. 5 u-velocity contour for xz plane in center of domain for training phase of Auto-Encoder

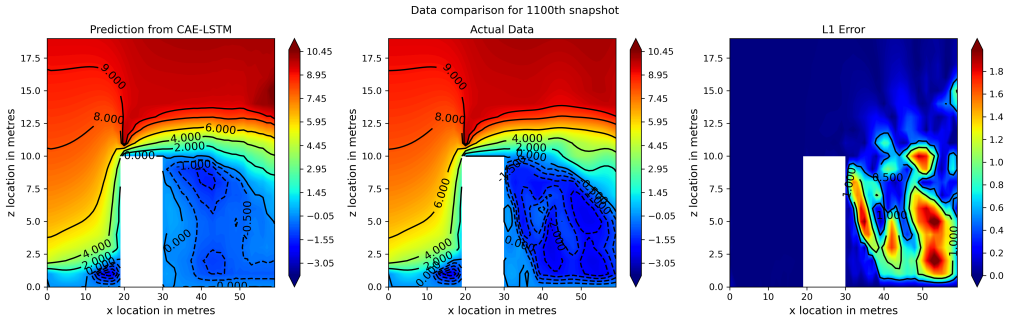
Unlike the CAE-LSTM case we use the Auto-Encoder network to predict for the next snapshot and this prediction is then used as input for the next prediction after that and so on. However, since we are using predictions are input for future snapshots, error accumulation might lead to worse predictions overtime compared to CAE-LSTM approach. This is reflected in the results depicted in the prediction phase as shown in figure 6.



(a) u-velocity contour for xz plane in center of domain for 700th snapshot



(b) u-velocity contour for xz plane in center of domain for 900th snapshot



(c) u-velocity contour for xz plane in center of domain for 1100th snapshot

Fig. 6 u-velocity contour for xz plane in center of domain for prediction phase of Auto-Encoder

IV. Conclusions and Future work

In this work we try to utilise Machine Learning based Reduced Order Models to generate ROMs for flow of wind around a building in a flow domain. We use techniques like Convolutional Auto-Encoders and LSTM networks to generate this ML-ROM. However this was pursued as a preliminary work and only 2 dimensional data snapshots were used to generate the 2D ROM. We aim to further extend this to the full 3D domain of interest. Furthermore since our aim was to demonstrate the method and its capability to generate efficient ROMs for our application, we use only the x-component of velocity for generating these models. This work could be futher extended especially for the CAE only approach, by using multiple previous snapshot data to predict the future snapshot, instead of just the current snapshot information. The predictions could also be improved over time with Data-Assimilation and incorporating physics of the flow into the Neural Network layers.

V. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1925147. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Some of the computing for this project was performed at the High-Performance Computing Center(HPCC) at Oklahoma State University supported in part through the National Science Foundation grant OAC-1531128. We would like to acknowledge high-performance computing support from Cheyenne[39] (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation. This work also used computational resources as part of the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC).

References

[1] de Oliveira Silva, L., de Mello Bandeira, R. A., and Campos, V. B. G., "The use of UAV and geographic information systems for facility location in a post-disaster scenario," *Transportation Research Procedia*, Vol. 27, 2017, pp. 1137–1145.

[2] Yao, H., Qin, R., and Chen, X., "Unmanned aerial vehicle for remote sensing applications—A review," *Remote Sensing*, Vol. 11, No. 12, 2019, p. 1443.

[3] Tsouros, D. C., Bibi, S., and Sarigiannidis, P. G., "A review on UAV-based applications for precision agriculture," *Information*, Vol. 10, No. 11, 2019, p. 349.

[4] Murphy, D. W., and Cycon, J., "Applications for mini VTOL UAV for law enforcement," *Sensors, C3I, information, and training technologies for law enforcement*, Vol. 3577, International Society for Optics and Photonics, 1999, pp. 35–43.

[5] Nex, F., and Remondino, F., "UAV for 3D mapping applications: a review," *Applied geomatics*, Vol. 6, No. 1, 2014, pp. 1–15.

[6] Grubesic, T. H., and Nelson, J. R., "UAS Platforms and Applications for Mapping and Urban Analysis," *UAVs and Urban Spatial Analysis*, Springer, 2020, pp. 13–29.

[7] León, C. D., "Drone Delivery? Amazon Moves Closer With F.A.A. Approval," , Aug 2020. URL `https://www.nytimes.com/2020/08/31/business/amazon-drone-delivery.html`.

[8] Mueller, E. R., Kopardekar, P. H., and Goodrich, K. H., "Enabling airspace integration for high-density on-demand mobility operations," *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 3086.

[9] Mohamed Salleh, M. F. B., and Low, K. H., "Concept of operations (ConOps) for traffic management of Unmanned Aircraft Systems (TM-UAS) in urban environment," *AIAA Information Systems-AIAA Infotech@ Aerospace*, 2017, p. 0223.

[10] Roth, M., "Review of atmospheric turbulence over cities," *Quarterly Journal of the Royal Meteorological Society*, Vol. 126, No. 564, 2000, pp. 941–990.

[11] Arnfield, A. J., "Two decades of urban climate research: a review of turbulence, exchanges of energy and water, and the urban heat island," *International Journal of Climatology: a Journal of the Royal Meteorological Society*, Vol. 23, No. 1, 2003, pp. 1–26.

[12] Onishi, A., Cao, X., Ito, T., Shi, F., and Imura, H., "Evaluating the potential for urban heat-island mitigation by greening parking lots," *Urban forestry & Urban greening*, Vol. 9, No. 4, 2010, pp. 323–332.

[13] Corburn, J., "Cities, climate change and urban heat island mitigation: Localising global environmental science," *Urban studies*, Vol. 46, No. 2, 2009, pp. 413–427.

[14] Tabassum, A., Vuppala, R. K., Bai, H., and Kara, K., "Variance Reduction of Quadcopter Trajectory Tracking in Turbulent Wind," *IFAC-PapersOnLine*, Vol. 54, No. 20, 2021, pp. 102–107.

[15] Siqueira, J. C. D. C., "Modeling of wind phenomena and analysis of their effects on UAV trajectory tracking performance," *Statler College of Engineering and Mineral Resources*, 2017.

[16] Raza, S. A., "Autonomous UAV control for low-altitude flight in an urban gust environment," Ph.D. thesis, Carleton University, 2015.

[17] Vuppala, R. K. S. S., and Kara, K., "Large-Eddy Simulation of Atmospheric Boundary-Layer Gusts for Small Unmanned Air Systems," *Bulletin of the American Physical Society*, 2020.

[18] Salim, S. M., Buccolieri, R., Chan, A., and Di Sabatino, S., "Numerical simulation of atmospheric pollutant dispersion in an urban street canyon: Comparison between RANS and LES," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 99, No. 2-3, 2011, pp. 103–113.

[19] Sutherland, M., "Urban wake field generation using LES for application to quadrotor flight," Ph.D. thesis, Carleton University, 2015.

[20] Lang, Y.-d., Malacina, A., Biegler, L. T., Munteanu, S., Madsen, J. I., and Zitney, S. E., "Reduced order model based on principal component analysis for process simulation and optimization," *Energy & Fuels*, Vol. 23, No. 3, 2009, pp. 1695–1706.

[21] Noack, B. R., Morzynski, M., and Tadmor, G., *Reduced-order modelling for flow control*, Vol. 528, Springer Science & Business Media, 2011.

[22] Xie, X., Mohebujjaman, M., Rebholz, L. G., and Iliescu, T., "Data-driven filtered reduced order modeling of fluid flows," *SIAM Journal on Scientific Computing*, Vol. 40, No. 3, 2018, pp. B834–B857.

[23] Rahman, S. M., Pawar, S., San, O., Rasheed, A., and Iliescu, T., "Nonintrusive reduced order modeling framework for quasigeostrophic turbulence," *Physical Review E*, Vol. 100, No. 5, 2019, p. 053306.

[24] Vuppala, R. K., and Kara, K., "A Novel Approach in Realistic Wind Data Generation for The Safe Operation of Small Unmanned Aerial Systems in Urban Environment," *AIAA AVIATION 2021 FORUM*, 2021, p. 2505.

[25] Landua, T. R., Vuppala, R. K. S. S., and Kara, K., "Investigation of Airflow around Buildings using Large Eddy Simulations for Unmanned Air Systems Applications," *AIAA SCITECH 2022 Forum*, 2022, p. 1688.

[26] Yildirim, O., San Tan, R., and Acharya, U. R., "An efficient compression of ECG signals using deep convolutional autoencoders," *Cognitive Systems Research*, Vol. 52, 2018, pp. 198–211.

[27] Holden, D., Saito, J., Komura, T., and Joyce, T., "Learning motion manifolds with convolutional autoencoders," *SIGGRAPH Asia 2015 technical briefs*, 2015, pp. 1–4.

[28] Vuppala, R. K. S. S., and Kara, K., "Realistic Wind Data Generation for Small Unmanned Air Systems in Urban Environment using Convolutional Autoencoders," *Bulletin of the American Physical Society*, Vol. 66, 2021.

[29] Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c., "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, 2015, pp. 802–810.

[30] Maronga, B., Gryschka, M., Heinze, R., Hoffmann, F., Kanani-Sühring, F., Keck, M., Ketelsen, K., Letzel, M. O., Sühring, M., and Raasch, S., "The Parallelized Large-Eddy Simulation Model (PALM) version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives," *Geoscientific Model Development*, Vol. 8, No. 8, 2015, pp. 2515–2551.

[31] Schumann, U., "Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli," *Journal of Computational Physics*, Vol. 18, No. 4, 1975, pp. 376–404. doi:https://doi.org/10.1016/0021-9991(75)90093-5, URL https://www.sciencedirect.com/science/article/pii/0021999175900935.

[32] Deardorff, J. W., "Stratocumulus-capped mixed layers derived from a three-dimensional model," *Boundary-Layer Meteorology*, Vol. 18, No. 4, 1980, pp. 495–527.

[33] Wyngaard, J., Peltier, L., and Khanna, S., "LES in the surface layer: Surface fluxes, scaling, and SGS modeling," *Journal of the atmospheric sciences*, Vol. 55, No. 10, 1998, pp. 1733–1754.

[34] Saiki, E. M., Moeng, C.-H., and Sullivan, P. P., "Large-eddy simulation of the stably stratified planetary boundary layer," *Boundary-Layer Meteorology*, Vol. 95, No. 1, 2000, pp. 1–30.

[35] Hochreiter, S., and Schmidhuber, J., "Long short-term memory," *Neural computation*, Vol. 9, No. 8, 1997, pp. 1735–1780.

[36] Franke, J., Hellsten, A., Schlünzen, K., and Carissimo, B., "Best practice guideline for the CFD simulation of flows in the urban environment-a summary," *11th Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes, Cambridge, UK, July 2007*, Cambridge Environmental Research Consultants, 2007.

[37] Murakami, S., and Mochida, A., "3-D numerical simulation of airflow around a cubic model by means of the k-e model," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 31, No. 2-3, 1988, pp. 283–303.

[38] Tutar, M., and Oguz, G., "Large eddy simulation of wind flow around parallel buildings with varying configurations," *Fluid Dynamics Research*, Vol. 31, No. 5-6, 2002, p. 289.

[39] Computational, and Laboratory, I. S., "Cheyenne: HPE/SGI ICE XA System (University Community Computing)," , 2017.