ON THE USE OF ECHO STATE NETWORKS IN VARIOUS CONFIGURATIONS TO PREDICT THE DYNAMICS OF ADVERSARIAL SWARMS

by

SOHAM GUPTA

JOHN BAKER, COMMITTEE CHAIR
SEMIH M.OLCMEN
RICHARD D.BRANAM
MUHAMMAD ALI ROB SHARIF
DANIEL J. FONSECA

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Aerospace Engineering
and Mechanics in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2021

ABSTRACT

Adversarial (competitive) swarms consist of two or more systems (each system consisting of a collection of individuals, interconnected agents) where the goals of each group are conflicting. This work aims to use an Echo State Network to predict the individual behavior of agents in two adversarial swarms and thereby develop an improved understanding of the dynamics of such systems. The current study was divided into three phases. An agent-based Adversarial swarm model was initially developed comprising of two competing swarms, the Attackers, and the Defenders, respectively. The Defender aimed to protect a point of interest in unbounded 2D Euclidean space called the Goal. In contrast, the Attacker's main task was to intercept the Goal while continually trying to evade the Defenders, which get attracted to it when they are in a certain vicinity of the Goal. The simulation was considered Semi-Hybrid as agent compromise, and goal compromise criteria were modeled to introduce realism as real-world engineering applications. The final system state was studied for the varied number of agents making up each swarm. The effectiveness of the Semi-Hybrid approach was validated by using Multiscale Entropy, which revealed a greater degree of randomness for the Defenders than Attackers. In the second investigation, two configurations were used to evaluate the use of Echo State Networks for predicting group dynamics for each swarm. Configuration 1 employed a single ESN, i.e., the patio-temporal data for all agents of an Adversarial Swarm model was used input. In configuration 2, two separate ESNs, in parallel, were used to predict Defender and Attacker swarm dynamics. It was concluded that the parallel ESN configuration was more effective in achieving qualitatively similar predictions of the dynamics for the Adversarial Swarms. In the final investigation, an instance of an ESN in a massively parallel framework was trained on individual spatio-temporal

data of every agent. The optimal hyperparameters obtained for every individual agent in the framework showed considerable variance that implied every agent in the Adversarial swarm reacted uniquely when a uniform stimulus was applied and thus reaffirmed the concept of individuality of agents in a swarm.

DEDICATION

This thesis is dedicated to my family, without whose unconditional support I would not have been where I am today.

LIST OF ABBREVIATIONS AND SYMBOLS

α	Leak rate
α_A	Constant of the self-propulsion force for Attackers.
α_D	Coefficient of the self-propulsion force for Defenders.
β	Regularization Coefficient
β_A	Coefficient of the Rayleigh friction force for Attackers.
β_D	Coefficient of the Rayleigh friction force for Defenders.
C_A	Depth of the potential well for pairwise scaled Morse potential for Attackers.
C_D	Depth of the potential well for pairwise scaled Morse potential for Defenders.
C_{DG}	The ratio of attraction and repulsive potential strength for scaled Morse potential between a Defender agent and the Goal.
k_{rep}	Constant for pairwise inter-swarm repulsive potential for Attackers and Defenders.
k_{att}	Constant for pairwise inter-swarm attraction potential for Defenders and Attackers.
k_{obj}	Constant for pairwise inter-swarm Goal potential between Attackers and the Goal.
l_A	The ratio of the repulsive to attractive length scales of Attacker agents.
l_D	The ratio of the repulsive to attractive length scales of Defender agents.
l_{DG}	The ratio of the length scale of attraction and repulsion pair for scaled Morse potential between a Defender agent and the Goal.
m_A	Mass of Attacker agents.
m_D	Mass of Defender agents.
n	Timestep.
$x(n)$	Neuron Activations.
$\tilde{x}(n)$	Neuron update.
W^{in}	Weight of input matrix.
W^{out}	Weight of output matrix.
AI	Artificial Intelligence
Config1	Configuration 1
Config2	Configuration 2
ESN	Echo State Network

RNN Recurrent Neural Network

RMSE Root Mean Square Error

ACKNOWLEDGEMENTS

I would like to acknowledge the faculty of The University of Alabama for their assistance and support with research and education both in and out of the classroom. I would also like to thank Dr. John Baker for his dedicated support in my research and education. I would also like to thank the additional members of my dissertation committee: Dr. Semih Olcmen, Dr. Richard Branam, Dr. Muhammad Sharif, and Dr. Daniel Fonseca.

I thank my present and ex-colleagues in the Advanced Space Technological Research and Applications (ASTRA) Group: Pradeep Sapkota, Shivank Shah, Chandler Nichols, and Morgan, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last five years.

I thank my parents, Mr. Rajkumar Gupta and Mrs. Rita Gupta, and my brother, Mr. Subham Gupta, for their faith in me and for allowing me to be as ambitious as I wanted. I am grateful for their unconditional support, love, and affection. I am also grateful to Ms. Swathi Babulal for her immense support in all avenues of life over the last five years of my Ph.D. journey. I am highly thankful to my cousin brother, Dr. Surojit Gupta, for his guidance, motivation, and immense knowledge. Special thanks to my cousin, sister, and brother-in-law Mrs. Aindrilla Gupta Niyogi and Mr. Shiladtiya Niyogi, for their continuous help and unconditional support throughout my time in the United States.My heart goes out to my friends Mr. Tanvir Sohail, Dr. Shuvodeep De, Mr. Sanjit Das, and countless others for keep me cheerful and happy throughout my last five years in Tuscaloosa, AL.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

All swarms are essentially complex systems generally characterized by nonlinear dynamics; an accurate physics-based model is imperative to the holistic understanding of the underlying dynamics. Swarms are a collection of independent, autonomous agents that are widespread in nature from ranging from ant colonies[1], flocks of birds[2], and schools of fishes[3]. Nature-inspired swarm models have been an active area of research for the past couple of decades. Simulating natural swarm behavior for engineering applications is inherently problematic as all-natural systems offer inherent flexibility and scalability, often difficult to attain using digital computers. Swarms form the basis of extensive modern-day engineering applications ranging from spacecrafts[4], UAVs[5], robots[6], and optimization algorithms[1].

Swarms have been historically modeled from a Lagrangian and an Eulerian (macroscopic) approach. In the former, the swarm agents and their interactions with the environment are modeled based on simple physics-based rules. In the former approach, the swarm behavior is generally analyzed based on collective attributes such a flock density. In the Lagrangian-based approach, the swarm agents and their interactions with the environment are modeled based on simple physics-based rules. Crag Reynolds's[7] seminal paper first introduced the idea of swarming where three simple rules, namely, 'attraction,' 'cohesion,' and 'velocity alignment, were used to model emergent swarming behaviors successfully. Reynolds[8] later introduced multiple other rules which enabled these agents to achieve a wide range of purposes. Huth[9]

introduced a computational model to mimic the behavior or school of fishes and compared synthetic data to real-world data. A discrete-time model on self-governed motion in a system was first introduced by Vicsek[10], where autonomous agents in a group coordinated with others by modified its velocity by adding to it the weight mean of the difference in velocities of other agents[11]. Vicsek's[10] kinematic model has been extensively studied by adding physics-based attributes and extensive modifications of the original model. Such changes include acceleration couplings for the self-propelled agents[12], the introduction of two types of interacting agents such as chasers and escapers having inertia, delay of communication, and noise[13], application of the original model to simulate indoor autonomous flying robots[14-16].

Swarm models discussed so far contains periodic boundary conditions that are typically inconsistent with nature as swarming in nature always occurs in free, unconstrained space. D'Orsogna[17, 18] proposed a dynamical systems-based swarming model using a generalized Morse potential coupled with Rayleigh friction force to study the structure of rotating flocks in free space. The primary force term in this model was derived from the Morse potential[19], which is repulsive at short ranges while being attractive at long ranges simultaneously; thus, it entails both close range repulsive and long-range attractive forces, which are essential components of any swarm preventing agent dispersion at the same time avoiding agent collisions. Vecil et al. [20] performed a 3D numerical simulation based on Orsoga's [18]model which, identified vital parameters that govern the nature of flocking behavior observed as such a rigid body rotation, clumps, milling, spheres, and dispersion. The consensus problem in the swarm made up of egalitarian agents was first explored by Cucker and Smale[21], which has been studied extensively by the addition of stochastic white noise [22], repulsive force[23], interparticle bonding force[24], and Rayleigh friction[25].

In nature, typically, two types of swarm interaction are observed-Adversarial and Symbiotic, wherein swarms either compete or cooperate over resource utilization. Symbiotic swarms found in the animal kingdom include multispecies group [26]hunting, were in different groups of species team up (cooperation) with each for hunting groups of prey. Adversarial swarms, which were the focus of the current study, are abundant in natural environments, such as groups of predators engaging with prey groups, and can be found both in aquatic and terrestrial environments. Most of such interactions take place for foraging purposes. Natural examples of Adversarial Swarms include groups of omnivorous Chimpanzees hunting groups of Red Colobus Monkeys[27] and groups of predator Lions hunting herds of Zebras[28].In the aquatic environment, a multispecies association of Dolphins with Seals and Dogfish for feeding schools of small fish[29], groups of Killer Whales, and a large number of Herring, where the former would force the latter to dive up by almost 150 meters[30], which would enable more effective foraging.

Historically, the Adversarial Swarm phenomena have been traditionally modeled as a predator-prey problem, explored by multiple groups of researchers, including ecologists, physicists, statisticians, and mathematicians. These models can be broadly classified into three types: kinematic, lattice-based, and dynamical. In addition to the computational models, few experimental studies have also been conducted in the recent past. In kinematic models, the interactions between agents are typically modeled as velocity terms. Angelani [31]investigated the collective predation in a simple agent-based model capable of reproducing animal movement patterns where the individual agents were modeled based on Vicsek's self-propelled[10] agents. Lin[32] used a self-propelled particle-based model to study the predation of bats on prey. In the lattice-based models, the computation domain is divided into uniform 2D grids or lattices, which

have 'states' associated with them, e.g., empty or filled. Notable lattice-based models include Kamimura et al.[33], where group chase and escape in a swarm modeled, and the study concluded the formation of highly self-organized spatial structures. Wang et al. [34]extended the predator-prey problem by adding a third species and considered the effects of stochastic vision; the study concluded a direct relationship between the predator's vision and the prey's extinction rate. Other notable works on swarms to swarm interaction includes Gaertner et al.[35], where an agent-based model based on the MASON library[36] was used to model the engagement between two groups of UAVs in 3D space. Strickland[37] studied swarm engagement during live experiments with two swarms of UAVs based on different pursuit and evasion strategies.

Dynamical swarm models are explicitly based on Newton's second law of motion, which offers accurate insights into the highly complex emergent behavior between the two swarms. Zhdankin et al. [38] studied the dynamics of a swarming predator-prey model, where each group's swarming was based on long-and short-range forces, and a non-conservative force was used to model the interaction term between the swarms. The study concluded the presence of Chaos, quasi-periodic, periodic behavior, and the existence of singularities. Kolon et al.[39] investigated the collision of two swarms made up of homogenous agents by considering the effect of delay in communication between various agents; the study demonstrated mutual swarm capturing during the interaction, ultimately leading to the milling[18] state of motion.

The current literature lacks the presence of an agent-based dynamical adversarial swarm model with explicitly defined 'inter' and 'intra' swarm forces. The first phase of the present study addressed this gap in the current literature by developing a physics-based dynamical adversarial homogenous swarm model with well-defined intra-swarm and inter-swarm forces. The Adversarial swarm model consisted of two distinct interacting swarms: The Attackers and the

Defenders, which had conflicting objectives in unbounded 2D Euclidean Space. The Defender's primary role was to protect the ‘Goal’ i, e a point of interest in 2D Euclidian space. In contrast, the Attacker's main objective was e to intercept the Goal while continually trying to evade the Defenders. The Defenders swarms always tried to protect the Goal by swarming around it and blocking any Attackers agent trying to reach the Goal. It was assumed that if an Attacker and Defender agents were very close to each other if the distance between them was less than a predefined criterion, they were assumed to have collided and were consequently arrested from participating in the simulation. The arrested agents became inactive for the remainder of the simulation. The simulation will be assumed to have a binary outcome or a final state, wherein either the Attacker or the Defenders emerge as the dominant swarm. The Attacker swarm was considered dominant if an agent in the swarm would successfully intercept the Goal during the simulation. If there were no remaining agents in the Defender swarm at any point in the simulation, the Attackers were be considered dominant. If the Defenders will successfully defend the Goal before the end of the simulation or if no Attackers were left in the simulation, the Defenders were considered the dominant swarm. If no agents were left in either of the swarms (i,e the agents compromise each other off in the engagement), the Defenders were considered dominant in the simulation as the Goal will be successfully protected from the predation of the Attacker swarm. The scenarios mentioned above formed the basis of the ‘Simulation Ending Criteria’ as presented in [40].

The model developed in the first phase of this study was essentially a multiagent dynamical system producing highly nonlinear time-series data. Traditional Artificial Neural Networks (ANNs) have been traditionally limited in predicting nonlinear transient times series data, which are highly non-stationary and non-cyclic in nature. ANNs have been historically combined with

other techniques such as NARMAX[41] to predict highly nonlinear chaotic systems such as the Lorentz system[42], Sunspot times series[43], and downhole pressure for a gas-lift oil well[44]. Elman et al.[45] introduced the idea of Recurrent Neural Networks, which were proven suitable for forecasting highly non-linear time-series data obtained from a dynamic system such as the Lorentz system[46]. However, training RNNs is a difficult task. The backpropagation through time (BPTT) method has been used to train RNNs successfully [47, 48]. Modified RNNs such as Long Short Term Memory (LSTM) have been successful in predicting high dimensional spatiotemporal systems in the short term[49], custom architectures such as Deep Neural Network(DNN) with convolutional LSTM[50] have also been successful. Oher state-of-the-art RNN architectures include Gated Recurrent Units (GRU-D), which has also been successful in predicting multivariate time-series with missing values by taking two representations of the missing patter, namely making and time interval. Random Recurrent Neural Network (rRNN) has also shown in success in predicting the periodic nonlinear Mackey Glass system[51]. One of the primary limitations with conventional RNN as indicated by Demiris [48] is the non-convergence in the training process due to existence of bifurcations. Slow convergence and high computational costs of RNNs also limit its utility for practical applications. However, a significant limitation of RNNs is the vanishing gradient problem[52] which severely limits its ability to learn long data sequences.

An alternative to gradient descent methods was thus proposed by Jager and Hass[53] as the Echo State Network(ESN) and by Maas[54] as the Liquid State Machine(LSM) in which only the synaptic connections form the RNN to output neurons were trained by learning. The idea of ESNs can be traced back to Neuroscience, Dominey et al.[55] presented a learning algorithm about sequence processing in mammalian brains, e.g., speech recognition in the human brain[55,

56], was the precursor of the actual algorithm of ESNs. ESNs and LSMs were unified into a common research topic known as the 'Reservoir Computing' (RC) [57, 58]. In an ESN, the main task is to construct an RNN with randomly generated weights. The randomly constructed complex nonlinear transformation of temporal data can be extracted from the output layer using simple techniques such as linear regression[59]. A great deal of art is needed to implement an ESN successfully. Several global hyperparameters must be tuned effectively. Since the first decade of the 2000s, RC-ESN has been successfully implemented in a multitude of domains, including speech recognition[60], robot control[61, 62], forecasting financial markets[63], natural language processing[64-67], Oil and natural gas sector such as pressure estimation in gas-lift oil wells[68], detrending of non-stationary fractal timeseries[69] and finally dynamical systems [53] such as the Mackey Glass system[51]. Recent research also reveals that hardware-based Reservoir computers are also possible based on FPGA arrays[70] and carbon nano tubes[71]. Compared to the current popular cloud computing trend, these computers can be more effective than traditional software and may be suited for 'edge computing'[70].

An ESN working effectively should satisfy have the echo state property. It is defined as a property where the effect of an earlier state or a previous input should vanish on the future state as time passes [72]. Mathematically, the echo state property is assumed to be maintained if the spectral radius value (the largest eigenvector) $\rho(\mathbf{W})$ of the reservoir weights is less than or equal to a value of 1. However, recent studies showed that the echo state property holds for $\rho(\mathbf{W}) \geq 1$ for nonzero inputs $\mathbf{u}(n)$, so $\rho(\mathbf{W})<1$ is not a necessary condition for the echo state property[73, 74].

The current state of the art revealed that ESNs have been very successful in predicting chaotic dynamic systems. ESNs in the existing literature have been found to use three

configurations while predicting chaotic behaviors, namely, observer mode (non-autonomous or predictive mode), where model free prediction is achieved by utilizing limited state variables [75], the generative (or autonomous) mode in which during the prediction the output of a previous timestep is fed as the input in the Reservoir [76-78] and finally custom ensemble methods where ESNs are used in conjunction with knowledge-based models[79]. The authors of the previously mentioned works successfully verified and validated these techniques in the simple Lorentz system[76], Lorentz-96 system[80], Kuramoto-Sivashinsky (KS) system[75, 78], the Rössler system[75], and dynamics of excitable media such as the Barkley model and the Bueno-Orovio-Cherry-Fenton model [81]. Hardware-based reservoir computers have also successfully predicted dynamical systems[70], such as the Mackey-Glass system. Krishnagopal[82] studied the effectiveness of reservoir computing for the separation of chaotic signals and concluded that their results were better than the Wiener filter obtained from the same training data.

Several studies have also been conducted to have a holistic understanding of a Reservoir computer's inherent dynamics, which would enable its effective use while predicting dynamic systems. Carroll[83] used an RC-ESN at the 'Edge of Chaos'[84, 85] region to perform predictions and concluded that it does not necessarily improve the performance. Carroll [86]also conducted studies on the dimension of Reservoir computers and concluded the increase of fraction dimension occurs inside the Reservoir with the increase of its dimension, which may adversely affect the performance of Reservoir Computer, Carroll[87] also conducted studies on the network structure of the RC-ESN. Zhang[88]conducted studies in the sensory phase coherence of two parallel reservoirs and concluded that short term prediction is possible, but parallel reservoirs are limited in sensing the collective dynamics of a coupled chaotic dynamic of the entire system in the long run.

In the second and the third phases of this investigation, ESN played a vital role in meeting the project objectives. Initial in-house studies showed that RC-ESN successfully predicted large-scale dynamical systems such as Large Eddy simulation of an incompressible turbulent round jet by implementing massively large-scale parallel reservoirs[89]. The second objective outlined in this study was achieved by using a high dimensional reservoir and two parallel high dimensional Reservoirs to predict the individual agent dynamics of the Attackers and Defenders, respectively. In this study's final objective, large-scale parallel reservoirs were used, wherein individual reservoirs were assigned to each agent of the Attackers and the Defenders, respectively. The respective agents were trained separately, and a comprehensive hyperparameter grid search was performed. The hyperparameters of the trained massively parallel reservoirs were studied statistically to explore the concept of 'individuality' in a complex adaptive system.

1.2 Motivation

Complex adaptive systems and machine learning are currently both highly active fields of research due to the widespread occurrence of such systems and to continued increases in computing power, respectively. Numerous examples of Adversarial Swarms' behavior, a type of complex adaptive system interaction, exist in nature (conflict between individual insect colonies/species, interactions between predator and prey species, etc.). Such behavior is also seen in human-related activities (team sports, military conflicts, political contests, etc.). An exhaustive literature survey has revealed that highly complex behavior can result from relatively simple rules governing the conduct of the individual agents comprising adversarial swarms. To date, research has focused on applying a single simple rule set to encode the response of all agents in a given swarm to a specific stimulus. Intuitively, it is known that all organisms (even those of the same species) do not react to identical stimuli uniformly. By using a step-by-step approach,

the variations in individual response affecting the behavior of a given swarm will be explored, in addition to its effect on the interaction between two Adversarial Swarms.

The current study will benefit society by advancing the understanding of a type of system that, while ubiquitous in the natural world, is increasingly being used to describe behavior observed in a myriad of disparate human applications. From the military use of drone swarms to cybersecurity to business analytics, insight gained from observed adversarial (competitive) swarms is being used to improve the efficiency and effectiveness of these and other complex systems. The physics-based Adversarial swarm model developed in this study represented an advancement that will ultimately serve as a benchmark for other researchers in the field. Successfully applying machine learning to predict complex agent-based swarming behavior also laid the groundwork for further study into a field that is rapidly changing.

1.3 Objectives

The current study contributed to the field of non-linear applied physics, machine learning, and complex adaptive systems in the following manner, respectively:

1. A novel physics-based adversarial swarm system will be modeled to study the nonlinear dynamics of the system. This model will further advance the understanding of the nature of dynamics observed during the interaction of two distinct swarms that are adversarial in nature. It will also advance the knowledge of the final outcome (i, e, the final state) of such interactions when studied across various populations of agents making up the two swarms, respectively. Also, it will validate the use of a novel semi-hybrid based assumption to similar agent-based models using Multiscale Entropy.
2. In the second part of this study, the prediction of the group dynamics of each swarm will be considered as a nonlinear time-series prediction problem using Echo State Networks (ESN),

a particular type of Recurrent Neural Network. This study further will further advance the field of applied machine learning by paving the way to predict high fidelity Multiagent Dynamical Complex Systems data.

3. In the final phase of the study, the prediction ‘individual’ agent dynamics will be considered. ESNs unique to each individual agent, respectively, will be used to predict the dynamics of individual agents. This investigation will study the effects of individuality. Time series data from respective individual agents will be used to train each agent's individual Echo State network. The impact of individuality will advance understanding how homogenous agents interact uniquely when a uniform external stimulus is applied in an environment. This study will advance the field of applied machine learning when applied to large-scale Multiagent Dynamical Complex Systems.

REFERENCES

1. Stanford, C. B. "The influence of chimpanzee predation on group size and anti-predator behaviour in red colobus monkeys," *Animal Behaviour* Vol. 49, No. 3, 1995, pp. 577-587.

2. Scheel, D. "Profitability, encounter rates, and prey choice of African lions," *Behavioral Ecology* Vol. 4, No. 1, 1993, pp. 90-97.

3. Vaughn, R. L., Shelton, D. E., Timm, L. L., Watson, L. A., and Würsig, B. "Dusky dolphin (Lagenorhynchus obscurus) feeding tactics and multi-species associations," *New Zealand Journal of Marine and Freshwater Research* Vol. 41, No. 4, 2007, pp. 391-400.

4. Selvi, V., and Umarani, R. "Comparative analysis of ant colony and particle swarm optimization techniques," *International Journal of Computer Applications* Vol. 5, No. 4, 2010, pp. 1-6.

5. Carere, C., Montanino, S., Moreschini, F., Zoratto, F., Chiarotti, F., Santucci, D., and Alleva, E. "Aerial flocking patterns of wintering starlings, Sturnus vulgaris, under different predation risk," *Animal Behaviour* Vol. 77, No. 1, 2009, pp. 101-107.

6. Calovi, D. S., Lopez, U., Ngo, S., Sire, C., Chaté, H., and Theraulaz, G. "Swarming, schooling, milling: phase diagram of a data-driven fish school model," *New journal of Physics* Vol. 16, No. 1, 2014, p. 015026

7. Perea, L., Gómez, G., and Elosegui, P. "Extension of the Cucker-Smale control law to space flight formations," *Journal of guidance, control, and dynamics* Vol. 32, No. 2, 2009, pp. 527-537.

8. Edwards, S. J. "Swarming on the Battlefield: past, Present, and Future." RAND NATIONAL DEFENSE RESEARCH INST SANTA MONICA CA, 2000.

9. Al-Rifaie, M. M., Aber, A., and Raisys, R. "Swarming robots and possible medical applications," 2011.

10. Reynolds, C. W. "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH computer graphics*. Vol. 21, ACM, 1987, pp. 25-34

11. Reynolds, C. W. "Steering behaviors for autonomous characters," *Game developers conference*. Vol. 1999, Citeseer, 1999, pp. 763-782.

12. Huth, A., and Wissel, C. "The simulation of fish schools in comparison with experimental data," *Ecological modelling* Vol. 75, 1994, pp. 135-146.

13. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. "Novel type of phase transition in a system of self-driven particles," *Physical review letters* Vol. 75, No. 6, 1995, p. 1226.

14. Janosov, M., Virágh, C., Vásárhelyi, G., and Vicsek, T. "Group chasing tactics: how to catch a faster prey," *New Journal of Physics* Vol. 19, No. 5, 2017, p. 053003.

15. Chuang, Y.-L., D'orsogna, M. R., Marthaler, D., Bertozzi, A. L., and Chayes, L. S. "State transitions and the continuum limit for a 2D interacting, self-propelled particle system," *Physica D: Nonlinear Phenomena* Vol. 232, No. 1, 2007, pp. 33-47.

16. D'Orsogna, M. R., Chuang, Y.-L., Bertozzi, A. L., and Chayes, L. S. "Self-propelled particles with soft-core interactions: patterns, stability, and collapse," *Physical review letters* Vol. 96, No. 10, 2006, p. 104302.

17. Mesa, A. D. S., Quesne, C., and Smirnov, Y. F. "Generalized Morse potential: symmetry and satellite potentials," *Journal of Physics A: Mathematical and General* Vol. 31, No. 1, 1998, p. 321.

18. Vecil, F., Lafitte, P., and Linares, J. R. "A numerical study of attraction/repulsion collective behavior models: 3D particle analyses and 1D kinetic simulations," *Physica D: Nonlinear Phenomena* Vol. 260, 2013, pp. 127-144.

19. Cucker, F., and Smale, S. "Emergent behavior in flocks," *IEEE Transactions on automatic control* Vol. 52, No. 5, 2007, pp. 852-862.

20. Ahn, S. M., and Ha, S.-Y. "Stochastic flocking dynamics of the Cucker–Smale model with multiplicative white noises," *Journal of Mathematical Physics* Vol. 51, No. 10, 2010, p. 103301.

21. Cucker, F., and Dong, J.-G. "Avoiding collisions in flocks," *IEEE Transactions on Automatic Control* Vol. 55, No. 5, 2010, pp. 1238-1243.

22. Park, J., Kim, H. J., and Ha, S.-Y. "Cucker-Smale flocking with inter-particle bonding forces," *IEEE Transactions on Automatic Control* Vol. 55, No. 11, 2010, pp. 2617-2623.

23. Ha, S.-Y., Ha, T., and Kim, J.-H. "Emergent behavior of a Cucker-Smale type particle model with nonlinear velocity couplings," *IEEE Transactions on Automatic Control* Vol. 55, No. 7, 2010, pp. 1679-1683.

24. Bailey, I., Myatt, J. P., and Wilson, A. M. "Group hunting within the Carnivora: physiological, cognitive and environmental influences on strategy and cooperation," *Behavioral Ecology and Sociobiology* Vol. 67, No. 1, 2013, pp. 1-17.

25. Nøttestad, L., Fernö, A., and Axelsen, B. E. "Digging in the deep: killer whales' advanced hunting tactic," *Polar Biology* Vol. 25, No. 12, 2002, pp. 939-941.

26. Angelani, L. "Collective predation and escape strategies," *Physical review letters* Vol. 109, No. 11, 2012, p. 118104.

27. Lin, Y., and Abaid, N. "Collective behavior and predation success in a predator-prey model inspired by hunting bats," *Physical Review E* Vol. 88, No. 6, 2013, p. 062724.

28. Kamimura, A., and Ohira, T. "Group chase and escape," *New Journal of Physics* Vol. 12, No. 5, 2010, p. 053013.

29. Wang, X., He, M., and Kang, Y. "A computational predator–prey model, pursuit–evasion behavior based on different range of vision," *Physica A: Statistical Mechanics and its Applications* Vol. 391, No. 3, 2012, pp. 664-672.

30. Gaertner, U. "UAV swarm tactics: an agent-based simulation and Markov process analysis." NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF OPERATIONS RESEARCH, 2013.

31. Luke, S. "Multiagent simulation and the MASON library," *George Mason University* Vol. 1, 2011.

32. Strickland, L., Day, M. A., DeMarco, K., Squires, E., and Pippin, C. "Responding to unmanned aerial swarm saturation attacks with autonomous counter-swarms," *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*. Vol. 10635, International Society for Optics and Photonics, 2018, p. 106350Y.

33. Zhdankin, V., and Sprott, J. "Simple predator-prey swarming model," *Physical Review E* Vol. 82, No. 5, 2010, p. 056209.

34. Kolon, C., and Schwartz, I. B. "The Dynamics of Interacting Swarms," *arXiv preprint arXiv:1803.08817*, 2018.

35. Woolley, J. W., Agarwal, P., and Baker, J. "Modeling and prediction of chaotic systems with artificial neural networks," *International journal for numerical methods in fluids* Vol. 63, No. 8, 2010, pp. 989-1004.

36. Zhang, G. P. "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing* Vol. 50, 2003, pp. 159-175.

37. Elman, J. L. "Finding structure in time," *Cognitive science* Vol. 14, No. 2, 1990, pp. 179-211.

38. Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., and Koumoutsakos, P. "Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics," *Neural Networks* Vol. 126, 2020, pp. 191-217.

39. Bishop, C. M. *Pattern recognition and machine learning*: springer, 2006.

40. Chatzis, S. P., and Demiris, Y. "Echo state Gaussian process," *IEEE Transactions on Neural Networks* Vol. 22, No. 9, 2011, pp. 1435-1445.

41. Vlachas, P. R., Byeon, W., Wan, Z. Y., Sapsis, T. P., and Koumoutsakos, P. "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* Vol. 474, No. 2213, 2018, p. 20170844.

42. Long, Y., She, X., and Mukhopadhyay, S. "HybridNet: integrating model-based and data-driven learning to predict evolution of dynamical systems," *arXiv preprint arXiv:1806.07439*, 2018.

43. Mackey, M. C., and Glass, L. "Oscillation and chaos in physiological control systems," *Science* Vol. 197, No. 4300, 1977, pp. 287-289.

44. Hochreiter, S. "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* Vol. 06, No. 02, 1998, pp. 107-116.

45. Jaeger, H., and Haas, H. "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *science* Vol. 304, No. 5667, 2004, pp. 78-80.

46. Maass, W., Natschläger, T., and Markram, H. "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation* Vol. 14, No. 11, 2002, pp. 2531-2560.

47. Dominey, P. F. "Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning," *Biological cybernetics* Vol. 73, No. 3, 1995, pp. 265-274.

48. Verstraeten, D., Schrauwen, B., d'Haene, M., and Stroobandt, D. "An experimental unification of reservoir computing methods," *Neural networks* Vol. 20, No. 3, 2007, pp. 391-403.

49. Schrauwen, B., Verstraeten, D., and Van Campenhout, J. "An overview of reservoir computing: theory, applications and implementations," *Proceedings of the 15th European Symposium on Artificial Neural Networks. p. 471-482 2007*. 2007, pp. 471-482.

50. Ferreira, A. A., Ludermir, T. B., and De Aquino, R. R. "An approach to reservoir computing design and training," *Expert systems with applications* Vol. 40, No. 10, 2013, pp. 4172-4182.

51. Skowronski, M. D., and Harris, J. G. "Automatic speech recognition using a predictive echo state network classifier," *Neural networks* Vol. 20, No. 3, 2007, pp. 414-423.

52. Antonik, P., Smerieri, A., Duport, F., Haelterman, M., and Massar, S. "FPGA implementation of reservoir computing with online learning," *24th Belgian-Dutch Conference on Machine Learning*. 2015.

53. Lin, X., Yang, Z., and Song, Y. "Short-term stock price prediction based on echo state networks," *Expert systems with applications* Vol. 36, No. 3, 2009, pp. 7313-7317.

54. Grezes, F. "Reservoir Computing." PhD thesis, The City University of New York, 2014.

55. Lukoševičius, M., Jaeger, H., and Schrauwen, B. "Reservoir computing trends," *KI-Künstliche Intelligenz* Vol. 26, No. 4, 2012, pp. 365-371.

56. Lu, Z., Pathak, J., Hunt, B., Girvan, M., Brockett, R., and Ott, E. "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 27, No. 4, 2017, p. 041102.

57. Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., and Ott, E. "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 27, No. 12, 2017, p. 121102.

58. Lu, Z., Hunt, B. R., and Ott, E. "Attractor reconstruction by machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 6, 2018, p. 061104.

59. Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B. R., Girvan, M., and Ott, E. "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 4, 2018, p. 041101.

60. Chattopadhyay, A., Hassanzadeh, P., Palem, K., and Subramanian, D. "Data-driven prediction of a multi-scale Lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ANN, and RNN-LSTM," *arXiv preprint arXiv:1906.08829*, 2019.

61. Zimmermann, R. S., and Parlitz, U. "Observing spatio-temporal dynamics of excitable media using reservoir computing," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 4, 2018, p. 043118.

62. Canaday, D., Griffith, A., and Gauthier, D. J. "Rapid time series prediction with a hardware-based reservoir computer," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 12, 2018, p. 123119.

63. Krishnagopal, S., Girvan, M., Ott, E., and Hunt, B. R. "Separation of chaotic signals by reservoir computing," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 2, 2020, p. 023123.

64. Carroll, T. L. "Mutual Information and the Edge of Chaos in Reservoir Computers," *arXiv preprint arXiv:1906.03186*, 2019.

65. Mitchell, M., Hraber, P., and Crutchfield, J. P. "Revisiting the edge of chaos: Evolving cellular automata to perform computations," *arXiv preprint adap-org/9303003*, 1993.

66. Crutchfield, J. P. "Between order and chaos," *Nature Physics* Vol. 8, No. 1, 2012, pp. 17-24

67. Carroll, T. L. "Dimension of reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 1, 2020, p. 013102.

68. Carroll, T. L., and Pecora, L. M. "Network structure effects in reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 29, No. 8, 2019, p. 083130.

69. Zhang, C., Jiang, J., Qu, S.-X., and Lai, Y.-C. "Predicting phase and sensing phase coherence in chaotic systems with machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 8, 2020, p. 083114.

70. Sapkota, P., Baker, J., and Gupta, S. "On the use of Reservoir Computing (Echo State Network) to predict the behaviour of an incompressible turbulent round jet," *Preprint*, 2020.

71. Gupta, S., and Baker, J. "Adversarial Swarms as Dynamical System," *Preprint*, 2020.

72. Chen, Y., and Yang, H. "Multiscale recurrence analysis of long-term nonlinear and nonstationary time series," *Chaos, Solitons & Fractals* Vol. 45, No. 7, 2012, pp. 978-987.

73. Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A. "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena* Vol. 16, No. 3, 1985, pp. 285-317.

74. Dalcín, L., Paz, R., Storti, M., and D'Elía, J. "MPI for Python: Performance improvements and MPI-2 extensions," *Journal of Parallel and Distributed Computing* Vol. 68, No. 5, 2008, pp. 655-662.

CHAPTER 2

ADVERSARIAL SWARMS AS DYNAMICAL SYSTEM

2.1 Abstract

An Adversarial Swarm model consists of two swarms that are interacting with each other in a competing manner. In the present study, an agent-based Adversarial swarm model is developed comprising of two competing swarms, the Attackers and the Defenders, respectively. The Defender's aim is to protect a point of interest in unbounded 2D Euclidean space referred to as the Goal. In contrast, the Attacker's main task is to intercept the Goal while continually trying to evade the Defenders, which get attracted to it when they are in a certain vicinity of the Goal, termed as the sphere of influence, essentially a circular perimeter. A semi-hybrid approach was considered for the simulation by adopting agent compromise and goal compromise criteria to introduce realism as per real-world engineering applications. The interaction of the two swarms was studied from a Dynamical systems perspective by changing the number of Agents making up each respective swarm. A total of 22 cases were studied for an ascending number of Attackers and descending number of Defender agents, starting with a population of 5 Attackers and 100 Defenders and ending with 100 Attackers agents and 5 Defenders agents. A Monte Carlo analysis is also conducted for initial randomized conditions for each run, respectively. The simulations were rigorously investigated for the presence of chaos by evaluating the Largest Lyapunov Exponent (LLE), implementing phase space reconstruction. Transient chaos was observed for some initial cases. The presence of chaotic behavior was also confirmed by plotting the Recurrence plot for some instances. The source of chaos in the system was observed to be induced by the passively constrained motion of the Defender agents around the Goal. Multiple

local equilibrium points existed for the Defenders in all the cases and some instances for the Attackers, indicating complex dynamics. LLEs for all the trials of the Monte Carlo analysis in all the cases revealed the presence of chaotic and non-chaotic solutions in each case, respectively, with most of the Defenders indicating chaotic behavior. Overall, the swarms exist in the 'Edge of Chaos,' displaying the existence of complex dynamical behavior. The final system state (the outcome of the interaction between the swarms in a particular simulation) was studied for all the cases, which indicated the presence of binary final states in some. Finally, to evaluate the nature of randomness and the effectiveness of the Semi-Hybrid approach, Multiscale Entropy is employed, which revealed a greater degree of randomness for the Defenders compared to Attackers. Overall, based on the Multiscale Entropy, the Semi-Hybrid approach of the simulation was considered successful as both the swarms exhibited an intermediate level of randomness.

2.2 Introduction

Swarms, which are a collection of individual autonomous agents, are omnipresent in nature ranging from ant colonies [1], flocks of birds [2], schools of fishes [3], and human crowds[90]. Swarming in nature provides many advantages, such as ensuring better survival chances against predators [91], collective foraging, and hunting [92]. Swarming behavior in nature has evolved over many thousands of years to be optimally adapted and prepared for the prevailing conditions in the environment. Bioinspired swarm models have been an active area of research for the past couple of decades. Mimicking natural swarm behavior for engineering applications is inherently problematic as all-natural systems offer inherent flexibility and scalability. The phenomenon of swarming has been exploited in modern-day engineering applications ranging from UAVs[93], optimization algorithms[1], robots [6], and spacecrafts[4].

All swarms are essentially complex systems characterized by nonlinear dynamics; an accurate physics-based model is essential to have a holistic understanding of the dynamics of swarming behavior. Historically physicists have modeled swarms from a Eulerian (Macroscopic) approach and a Lagrangian (Individual, agent-based) modeling approach. In the former approach, the swarm behavior is generally analyzed based on collective attributes such a flock density. In the Lagrangian-based approach, the swarm agents and their interactions with the environment are modeled based on simple physics-based rules. Complex emergent behavior is observed, which is as a result of simple interactions at the agent level. Craig Reynold's s [7]seminal paper introduced the idea of swarm modeling, which showed that three simple rules could achieve emergent swarming behavior, namely - attraction, collision, and velocity alignment. Reynolds [8]subsequently added further rules called 'steering behaviors' to diversify swarming behaviors to achieve specific objectives. Huth et al. [9]introduced a model to describe fishes' behavior schools and compared simulated motion produced from his model with real-world data. Vicsek et al. [10]proposed a discrete-time model on self-governed motion in a system of particles. In Vicsek's model, every autonomous agent modified its velocity by adding the weighted mean of the difference in other agents' velocities. Vicsek's [10]computer simulations showed agents approaching the same velocity with time; that is, the particles behaved like swarms having coordinated motion. Vicsek et al.[10] has been studied extensively, and several variations were proposed, including the addition of inertia, time-delay, and noise[13]; all of Vicsek's models are kinematic. Viscido et al. [91]studied population size and the neighbors on the emergent properties, including polarity, edges, and distinct shapes.

The swarm models discussed so far consider mostly periodic boundary conditions, which are generally inconsistent with nature as most natural swarms flock in free space without any

explicit boundary constraints. D'Orsogna et al. [17, 18] proposed a swarming model using a generalized Morse potential coupled with Rayleigh friction force to study the structure of rotating flocks in free space. Forces derived from the pairwise Morse potential are repulsive at short ranges while being attractive at long ranges simultaneously; thus, it entails both close range repulsive and long-range attractive forces, which are essential components of any swarm. The Morse potential[19] is an integral component of the current work and is discussed in detail at a later stage. Vecil et al. [20] performed a 3D numerical simulation based on Orsoga's [18]model which, identified vital parameters that govern the nature of flocking behavior observed as such a rigid body rotation, clumps, milling, spheres, and dispersion. Cucker and Smale [21]introduced a simple dynamical system to tackle the consensus problem in a leaderless non-hierarchical swarm's constituent agents. The Cucker-Samle[21] type model has been studied extensively, and the model has been developed further by adding extensive attributes as the addition of repulsive force[23], addition to stochastic white noise[22], interparticle bonding force[24], and Rayleigh friction[25]. All the models mentioned above are dynamic in nature and consider unconstrained swarming in free space.

Adversarial swarms, the focus of the current study, are abundant natural environments, such as groups of prey engaging with groups of prey, both in aquatic and terrestrial environments for foraging purposes. In nature, swarms often interact with other swarms; interactions can be symbiotic or adversarial, depending on the nature of the swarms involved. Examples of symbiotic swarms found in the animal kingdom include multispecies group [24]hunting, where different groups of species team up (cooperation) with each other for hunting groups of prey. Terrestrial examples of Adversarial swarms include groups of omnivorous Chimpanzees hunting groups of Red Colobus Monkeys[27], groups of predator Lions hunting herds of Zebras[28], etc., in the

aquatic environment, a multispecies association of Dolphins with Seals and Dogfish for feeding schools of small Fish[29], groups of Killer Whales and tens of tons of Herring, where the former would force the later to dive up by almost 150 meters[30]to indulge in more effective foraging.

Many Adversarial Swarm models have been adopted over the years in which physicists have traditionally modeled the adversarial swarms phenomenon as a predator-prey problem. Multiple research communities have explored these models, such as ecologists, physicists, statisticians, and mathematicians. The current literature reveals three main types of models - kinematic, lattice-based, and dynamical models. In addition to the computational models, few experimental studies have also been conducted in the recent past. In kinematic models, the interactions between agents are typically modeled as velocity terms. Angelani [31]investigated the collective predation in a simple agent-based model capable of reproducing animal movement patterns where the individual agents were modeled based on Vicsek's self-propelled[10] agents. Lin[32] used a self-propelled particle-based model to study the predation of bats on prey. The computation domain is divided into uniform 2D grids or lattices in the lattice-based models, which have 'states' associated with them, e.g., empty or filled. Notable lattice-based models include Kamimura eta al.[33], where group chase and escape in the lattice-based swarm were modeled, and the study concluded the formation of highly self-organized spatial structures. Wang et al. [34]extended the predator-prey problem by adding a third species and considered the effects of stochastic vision, concluding a direct relationship between the predator's vision and the prey's extinction rate. Other notable works on swarms to swarm interaction includes Gaertner et al.[35], where an agent-based model based on the MASON library[36] was used to model the engagement between two groups of UAVs in 3D space. Strickland[37] studied swarm engagement during live experiments with two swarms of UAVs based on different pursuit and

evasion strategies.

Dynamical system-based Adversarial swarm models are explicitly based on Newton's second law of motion which, can provide accurate insights into the highly complex emergent behavior arising between the interaction of two swarms. Zhdankin et al. [38] studied the dynamics of a swarming predator-prey model, where each group's swarming was based on long- and short-range forces, and a non-conservative force was used to model the interaction term between the swarms. The study concluded the presence of chaos, quasi-periodic, periodic behavior, and the existence of singularities. Kolon et al.[39] investigated the collision of two swarms made up of homogenous agents by considering the effect of delay in communication between various agents; the study demonstrated mutual swarm capturing during the interaction, ultimately leading to a 'milling' [18] state of motion.

Overall, the current literature available in the public domain lacks the presence of an Agent-based dynamical Adversarial Swarm model with explicitly defined 'inter' and 'intra' swarm forces, which are based on widely-accepted physics-based potential functions that have been historically successful to model the behavior of simple swarming system capable of producing highly emergent behavior. This gap in the current literature was addressed by developing a physics-based dynamical adversarial homogenous swarm model with very well-defined intra-swarm and inter-swarm forces. The current literature available lacks an Agent-based dynamical Adversarial swarm model with explicitly defined 'inter' and 'intra' swarm forces based on physics-based potential functions as many similar models in the literature have force terms that are 'ad-hoc' based that are sometimes exclusively dependent on empirical observations. Also, most of these models disregard the mass or inertia, which are an integral component of such dynamical systems. Moreover, the current model also adopts a Semi-Hybrid

approach to model collision and agent compromise. The model consists of two distinct interacting swarms: The Attackers and the Defenders, with conflicting objectives in unbounded 2D Euclidean Space. The Defenders role is to protect the Goal, a point of interest in 2D Euclidian space. In contrast, the Attackers' primary objective is to intercept the Goal while continually evading the defenders. The Defender swarm tries to protect the Goal by swarming around it and blocking any Attackers agent trying to reach the Goal. It is assumed that if an Attacker and Defender agents are very close to each other, or, in other words, if the distance between them is less than a predefined criterion, they are considered to collide with each other and are consequently arrested from participating in the simulation. As a result of the interaction, the arrested agents' count may not always be binary. The arrested agents become inactive for the reaming time of the simulation. The simulation is assumed to have a binary outcome or a final state, wherein either the Attacker or the Defenders emerge dominant. The Attacker swarm is considered dominant if an agent in the swarm can successfully intercept the Goal during the simulation. If there are no remaining agents in the Defender swarm at any point in the simulation, then the Attackers are also considered dominant. If the Defenders can successfully defend the Goal before the end of the simulation or if no Attackers are left in the simulation, the Defenders are considered as the dominant swarm. If at any time during a simulation, no agents are left in either of the swarms (the agents compromise each other off in the engagement), the Defenders are dominant in the simulation as the Goal has been successfully protected from the predation of the Attacker swarm. Thus, the simulations can be referred to as being Semi-Hybrid in nature, where agent compromise and goal interception are modeled per practical scenarios that may arise in engineering applications. To the authors' knowledge, the current literature lacks the presence of such Semi-Hybrid models.

In the current study, the dynamical system-based Adversarial Swarm Model simulation is carried out on a robustly developed computational platform in C++ to solve Newton's second law problem. Before running the simulations, a rigorous convergence study is carried out to determine the optimal timestep. This aspect is often missing in many ad-hoc based modeling approaches found in the current literature. The nonlinear time series data obtained from the simulation uses a multitude of tools that include time-series plots, attractor plots, and Largest Lyapunov Exponent (LLE) and Recurrence Analysis. The system is strongly investigated for the presence of chaos. As a vital parameter of the system, the number of Attackers and the Defender agents making up each swarm is varied to study the simulation's final state. The temporal data obtained from the simulation is analyzed using Multiscale Entropy to evaluate the degree of randomness in the system as the data comes from a semi-hybrid simulation. This approach will help rule out whether the current semi-hybrid approach is successful, and it is also hypothesized that the time-series data obtained from the simulations may have an intermediate level of randomness. The application of Multiscale entropy is integral to the current work as to the author's knowledge it has never been used to evaluate the success of a semi-hybrid simulation approach build by an agent-based modeling approach.

The intellectual contribution of this paper can thus be summarized as follows:

- Development of a generic physics-based Adversarial Swarm using an Agent-based modeling approach.
- Model was semi-hybrid in nature as agent compromise, and simulation end criteria were built into the model
- Model was solved using a rigorous numerical approach, unlike similar models
- The model was strongly studied from dynamical systems point to view to provide a clear

understanding of the physics of interacting swarm behavior

- Multi-scale entropy was used to study the degree of randomness in the data obtained to determine the success of the semi-hybrid approach assumed.

This paper is organized as follows- section 2.3 discusses the numerical model and the computational method used for solving the governing equations. Section 2.4 briefly describes the nonlinear time series analysis techniques used, followed by the results and discussion in section 2.5. The final section 2.6 comprises the conclusion.

2.3 Numerical Model and Parameter Descriptions

A physics-based agent-based model was derived for studying the dynamics of two interacting adversarial swarms: The Attacker Swarm and the Defender Swarm (hence, referred to as 'Attackers' and 'Defenders' respectively). The agents had conflicting objectives; the Defenders protected a point of interest in unbounded 2D Euclidean space by swarming around the Goal along a sphere of influence. In contrast, the Attackers' main task was to intercept the Goal while constantly trying to evade the Defenders, who actively chased the former in a perimeter around the Goal or a sphere of influence. The individual swarms in the swarm system are modeled based on a Lagrangian-based approach having primarily two types of forces- 'inter' and 'intra' swarm forces; the inter-forces are used to model the interaction between the agents of the adversarial swarms, respectively. The intra-forces were used to model the forces between members of the same swarm. Each swarm can be generalized as a collection of N agents in a 2-Dimensional space with position and velocity vectors. The governing equation describing the dynamics of the two interacting swarms- the Attackers and the Defenders were derived based on Newton's second law of motion and are given by the following equations:

$$\ddot{\vec{X}}_{A_i} = \dot{\vec{V}}_{A_i} = \frac{1}{m_{A_i}}\left(\sum_{\substack{A_i=1 \\ A_i \neq A_k}}^{N_A} -\left(\vec{\nabla}\varphi_{A,ik}\right) + \sum_{A_i=1}^{N_D} -\vec{\nabla}\left(k_{rep} r_{ij}^{-1}\right) - \vec{\nabla}(-k_{obj} r_{iG}^{2}) + (\alpha_A - \beta_A |\vec{V}_{A_i}|^2)\vec{V}_{A_i}\right. \tag{2.1}$$

$$\dot{\vec{X}}_{A_i} = \vec{V}_{A_i} \tag{2.2}$$

$$\ddot{\vec{X}}_{Dj} = \dot{\vec{V}}_{Dj} = \frac{1}{m_{Dj}}\left(\sum_{\substack{Dj=1 \\ Dj \neq Dh}}^{N_D} -(\vec{\nabla}\varphi_{D,jh}) + \sum_{Dj=1}^{N_A} -\vec{\nabla}(-k_{att} r_{ji}^{-1}) - \vec{\nabla}(\varphi_{jG}) + (\alpha_D - \beta_D |\vec{V}_{Dj}|^2)\vec{V}_{Dj}\right. \tag{2.3}$$

$$\dot{\vec{X}}_{Dj} = \vec{V}_{Di} \tag{2.4}$$

Eqns. (2.1-2.4) are the principal equations for the Adversarial Swarm model that, subject to given initial conditions$\vec{V}_{A_i}(t=0), X_{A_i}(t=0), \vec{V}_{D_j}(t=0), X_{D_j}(t=0)$ of individual agents in the respective swarms are known.

The first terms in Eqns. (2.1) and (2..3) respectively are intra-swarm forces that are modeled based on the scaled Morse[36] potential. The gradient of the potential was used to derive the intra-swarm Morse force; the following equation gives a generalized scaled Morse potential.

$$\varphi_{ij} = C\exp(-\frac{|r_{ik}|}{l}) - \exp(-\frac{|r_{ik}|}{l}) \tag{2.5}$$

In Eqn. (2.5), C defines the depth of the repulsive potential well, and l is a constant used to relate the ratio of the repulsive to attractive length scales. The forces obtained from the scaled Morse potential are responsible for the swarming of agents in the Attackers and Defenders. The typical intra-swarm Morse force scenario is typically C > 1 and l > 1, which means that the repulsive component only acts at close ranges. However, the attractive component works at long ranges only, thereby preventing the dispersion of agents making up a swarm, respectively. From

D'Orsoga et al.[18], it is known a criterion called H stability is necessary, under which if the number of particles increases, the Morse force guarantees that the bounding of the swarm as the number of particles increases. The H stability can be achieved by imposing the condition $Cl^2 > 1$, lest it might lead to 'catastrophic' behavior [17]. The Morse force constants in both the Attackers and Defenders are chosen by imposing the H stability criteria. The Morse potential is an ideal choice as it would implicitly impose limits on the range of sense when compared to using hard-coded cut-off distances as biological entities and engineered artificial agents would have an implicitly limited range of vision or sensing capability.

The second term in Eqns. (2.1) and (2.3) is an attractive and repulsive potential. The Defenders stop the Attackers from fulfilling their primary objective: the interception of the Goal. The Defenders agents use an attractive force derived from the second term's attractive potential in Eqn. (2.3) to intercept the Attacker agents. The Attacker agents try continually to evade the Defender agents utilizing a repulsive force derived from the second term of Eqn. (2.1). The attractive and repulsive potential found in Eqns. (2.1) and (2.3) are derived from a generalized obtained from Espitia et al.[94]. This potential serves a dual purpose, as it can be used to derive attractive force and repulsive force by merely changing its sign. The following equation can compactly express the attractive/repulsive potential.

$$\varphi_A = \pm k_{rep/att} r_{ij}^{-1} \tag{2.6}$$

Where the first term in Eqn. (2.6) is a positive constant term (with suffix 'rep') if a repulsive force is derived and negative (with suffix 'att') if an attractive force is derived. The Defender agents only get attracted to the Attackers' agents inside the sphere of influence, as depicted by the circle in Fig.2.1. It is a necessary step, which ensures that the Defenders do not veer off too far from the Goal, as the Defender's main aim is to protect the Goal from the Attackers. The

following equation defines the attractive force between the Defender agents and the Attackers agents inside the sphere of influence:

$$\vec{F}_{DA,A} = -\nabla(-k_{att} r_{ij}^{-1}) \text{ if } r_{ij} \leq R_{th} \tag{2.7}$$

$$\vec{F}_{DA,A} = 0 \text{ if } r_{ij} > R_{th} \tag{2.8}$$

Where $R_{th}$is a threshold radius around the Goal. The sphere of influence also contributes to making the system semi-hybrid in nature. The attractive and repulsive forces are inversely proportional to the competing agents' distance, ensuring that the attraction and repulsion are strongest between the closest agents. The attraction and the repulsion forces are cut off using local distance thresholds, beyond which they would essentially be constants equal to the attraction/repulsive force at the threshold distances. The threshold distances prevent these forces from getting too large during the very close interaction of Attacker and Defender agents, thereby warranting a tractable computation. These distance thresholds are assumed to be the same for both classes of agents to ensure fair competition.

The third term in equation (2.1) is an attractive potential [94], which is used to obtain an attractive goal force between the Attackers and the Goal. The attraction force derived from this potential is a linear force that increases as the distance between an Attacker agent and the Goal increases and vice versa.

Where $R_{th}$is a threshold radius around the Goal. The sphere of influence also contributes to making the system semi-hybrid in nature. The attractive and repulsive forces are inversely proportional to the competing agents' distance, ensuring that the attraction and repulsion are strongest between the closest agents. The attraction and the repulsion forces are cut off using local distance thresholds, beyond which they would essentially be constants equal to the attraction/repulsive force at the threshold distances. The threshold distances prevent these forces

from getting too large during the very close interaction of Attacker and Defender agents, thereby warranting a tractable computation. These distance thresholds are assumed to be the same for both classes of agents to ensure fair competition.

The third term in equation (2.1) is an attractive potential [94], which is used to obtain an attractive goal force between the Attackers and the Goal. The attraction force derived from this potential is a linear force that increases as the distance between an Attacker agent and the Goal increases and vice versa.

The third term in Eqn. (2.3) is also derived from a scaled Morse potential [39] (similar to the potential used for deriving the intra-swarm forces) is a force between the Defenders and the Goal, which has a repulsive as well as an attractive component. This force prevents the Defender agents from wandering too far off from the Goal. The following equation gives the scaled Morse potential between the Goal and a Defender agent.

$$\varphi_{jG} = C_{DG}\exp(-\frac{|r_{jG}|}{l_{DG}}) - \exp(-|r_{jG}|) \tag{2.9}$$

Where r_{jG} is the distance between a Defender agent and the Goal. C_{DG} and l_{DG} are constants that can control the composition of the attractive and repulsive force between the Defenders and the Goal. As discussed in the preceding section, the H stability condition ($Cl^2 > 1$)is imposed to avoid a 'catastrophic' scenario[17, 18].

The last terms in the Eqns. 2.1 and 2.4 is the self-propelling and frictional force term, which is based on Rayleigh's friction. The Rayleigh friction force is derived from the Rayleigh Dissipation function [95]. It is a nonlinear damping term with self-acceleration and friction mechanisms, which drive all the particles to an equilibrium speed of α/β [15,36,37]. The Rayleigh Friction force is a velocity-based force that is non-conservative and is given by:

$$\vec{F}_{A,SF,i} = (\alpha_A - \beta_A|\vec{V}_{A,i}|^2)\vec{V}_{A,i} \tag{2.10}$$

$$\vec{F}_{D,SF,i} = (\alpha_D - \beta_D|\vec{V}_{D,i}|^2)\vec{V}_{D,i} \tag{2.11}$$

2.3.1 Initial Conditions, Verification, Validation, Uncertainty Quantification, and Parameter Selection/description:

Eqs. (2.1)-(2.4) are numerically integrated using a customized 4th order Runge Kutta explicit solver [96] for Newton's Second law of motion. The agents are initialized within the square domain with four distinct quadrants, as shown in Fig 2.1. The Goal is located at the center of the fourth quadrant (0.5, -0.5); in this study, the Goal's location is considered fixed for all numerical experiments.

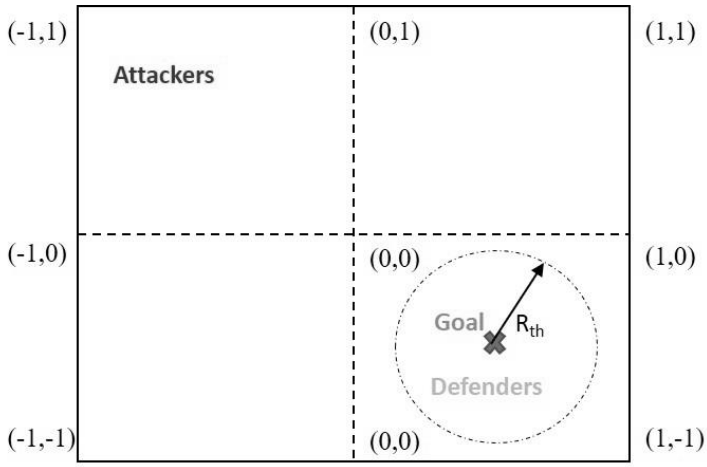

Figure 2.1. Illustration of the domain for simulation.

The dashed circle around the Goal represents the sphere of influence found in Eqn (2.7) and (2.8), fixed at 0.508 for this study. The Defender positions are initialized randomly inside the sphere of influence, in the fourth quadrant, using a low discrepancy Sobol's sequence[97], the velocities are randomly chosen between ±0.1. The Attackers are also randomly generated inside the second quadrant using the same technique, and the velocities were also randomly selected between ±0.1. Low discrepancy Sobol's sequence is chosen to ensure minimal overlap between

the initial position vector of agents making up their respective swarms.

Table 2.1: Various Key Parameters used in the numerical model

Constants	Values	Description
C_A	1.32	Depth of the potential well for pairwise scaled Morse potential for Attackers.
l_A	1.50	The ratio of the repulsive to attractive length scales of Attacker agents.
C_D	1.38	Depth of the potential well for pairwise scaled Morse potential for Defenders.
l_D	2.00	The ratio of the repulsive to attractive length scales of Defender agents
k_{rep}	0.054	Constant for pairwise inter-swarm repulsive potential between Attackers and Defenders.
k_{att}	2.82	Constant for pairwise inter-swarm attraction potential between Defenders and Attackers.
k_{obj}	10	Constant for pairwise inter-swarm Goal potential between Attackers and the Goal.
C_{DG}	10.03	The ratio of attraction and repulsive potential strength for scaled Morse potential between a Defender agent and the Goal.
l_{DG}	0.5	The ratio of the length scale of attraction and repulsion pair for scaled Morse potential between a Defender agent and the Goal.
α_A	1	Constant of the self-propulsion force for Attackers.
β_A	1	Coefficient of the Rayleigh friction force for Attackers.
α_D	1	Coefficient of the self-propulsion force for Defenders.
β_D	1	Coefficient of the Rayleigh friction force for Defenders.
R_{th}	0.508	Influence radius for Defenders.
m_A	1	Mass of Attacker agents.
m_D	1	Mass of Defender agents
Goal breach criteria	1E-4	Threshold distance between Attacker agents and the Goal for considering the Goal to be breached.
Agent compromise criteria	1E-4	Threshold distance between individual Attacker agents and Defender agents to be considered collided (hence dead).
Repulsive Force Local Cut-off for Attacker	0.05	The minimum distance beyond which the repulsive force between an Attacker and a Defender agent is treated as constant.
Attraction Force Local Cut-off for Defenders	0.05	The minimum distance beyond which the attractive force between a Defender and an Attacker agent is treated as constant.

The Runge Kutta solver is first verified by testing it against a trivial mass damper system. The validation of the custom solver developed for this problem is achieved by numerically evaluating the order of accuracy. In time, a comprehensive grid independence study in time is carried to evaluate the optimum timestep for solving the underlying governing equations for each

agent, respectively, which is depicted in the subsequent section. A relative error threshold of 1% was considered in the position vectors for both the agents to perform the grid independence analysis in time.

The various constants' optimal values were obtained using a combination of trial and error, educated guess, and visual inspection (rendering of the simulation) over multiple random and non-random numerical experiments. Table 2.1 describes the model constants and values (Note: Suffix A denotes Attackers and suffix D represents Defenders). If the distance between two or more different agents is equal or less than the 'Agent compromise criteria', they are considered to have collided and are hence arrested from the simulation. The arrested agents become inactive for the remainder of the simulation. If the distance between any Attacker agent and the Goal is less than the 'Goal breach criteria', the Goal is considered breached. As stated in the introductory section, if any Attacker agent can successfully intercept the Goal, Attackers dominate the simulation. If the Defenders can defend the Goal before the end of the simulation or if no Attackers are left in the simulation, then the Defenders is considered the dominant swarm. If no Attacker or Defender agents are left at any point in the simulation, in that case, the Defenders are considered dominant as the Goal has been successfully protected from the Attacker's predation. The various threshold distances discussed so far can be found in Table 2.1.

2.4 Results and Discussion

All the simulations were carried out by developing custom software written in C++ and Python, which were run on the University of Alabama High-Performance Computing Network and a Desktop Computer having Intel® Core™ i7-9700 Processor,64 GB RAM, and 1.25 TB of storage space. The simulations were carried out for different populations of Attackers and Defenders to study the dynamics and the outcome (final state) of the interacting adversarial swarms. The interaction between the two adversarial swarms was analyzed with respect to an ascending ratio of the number of Attackers to Defenders. The maximum number of Attackers and Defenders agents in this study was limited by the computational resources available, which was capped at a maximum of 100 for each case. In the initial simulation, the number of attackers

Table 2.2: Case Matrix

Case	NA	ND	NA/ND	Runs	Timestep	Max Simulation Time
1	5	100	0.05	1000	5.00E-05	100
2	7	97	0.07	1000	2.50E-05	100
3	10	95	0.10	1000	2.50E-05	100
4	15	90	0.16	1000	2.50E-05	100
5	20	85	0.23	1000	2.50E-05	100
6	25	80	0.31	1000	2.50E-05	100
7	30	75	0.4	1000	2.50E-05	100
8	35	70	0.5	1000	2.50E-05	100
9	40	65	0.61	1000	2.50E-05	100
10	45	60	0.75	1000	2.50E-05	100
11	50	55	0.90	1000	2.50E-05	100
12	50	50	1	1000	2.50E-05	100
13	55	50	1.1	1000	2.50E-05	100
14	60	45	1.33	1000	2.50E-05	100
15	65	40	1.62	1000	2.50E-05	100
16	70	35	2	1000	2.50E-05	100
17	75	30	2.5	1000	1.00E-05	100
18	80	25	3.2	1000	5.00E-06	100
19	85	20	4.25	1000	5.00E-06	100
20	90	15	6	1000	5.00E-06	100
21	95	10	9.5	1000	5.00E-06	100
22	100	5	20	1000	2.00E-06	100

was 5; the number of Defenders chosen was 100(case #1); for every subsequent case excepting case#2, the Attackers were increased by 5, and the Defenders reduced by 5; which was continued until there were 100 Attackers and 5 Defenders left.

A total of 22 cases were studied in total, as shown in Table 2.2. All simulations had random initial conditions for position and velocity; a Monte Carlo study was carried out to understand each case's outcome holistically. Every case was run 1000 times, determined by the cumulative average of each run's total time. The cumulative average had a maximum change of 0.2% at the end of 1000 runs for each case, respectively. The amount of computational resources available also limited the total number of runs for each case, respectively. Initial random conditions were chosen for each run to implicitly introduce noise into the study of the overall system dynamics of the system. It was anticipated that the random initial conditions would cause the system to in various dynamical regimes.

The simulations were studied from a swarm to a swarm interaction perspective. The center of mass time-series of the swarm was found out, averaging the x and the y coordinates of the position vectors with respect to the total number of active agents at every timestep. The center of the momentum of the swarms was found out by averaging the x and the y components of the velocity vector with respect to the total number of active agents at every timestep. In the preceding calculations, the individual agents' mass making up the respective swarms is considered unity

The largest Lyapunov Exponents (LLEs) of the center of mass time-series for both agents were obtained using Wolf's algorithm [98]by implementing phase space reconstruction by evaluating the minimum embedding dimension and time lag from Chen et al.[99]. Lyapunov exponent is a useful tool for determining the presence of chaos. Lyapunov exponents were found

out for every trial run in each case, respectively. The following equation gives the Largest Lyapunov Exponent (LLE):

$$\Lambda_1 = \frac{1}{M_{LE} t_{evol}} \sum_{k=0}^{M} \ln \frac{L_{evol}^{(i)}}{L_o^{(i)}} \tag{2.12}$$

The Euclidean distance between the initial point and its nearest neighbor $t = 0$ is given by $L_o$which evolves into L_{evol} after $t = t_{evol}$and $M_{LE}$is the minimum embedding dimension of the reconstructed phase space.

Recurrence plots(RP)[100] indicate whether a dynamical system is periodic or chaotic in nature are essentially two-dimensional representation techniques of a symmetric binary square matrix that brings out distance correlations in a time-series. Binary mapping is used to construct the symmetric binary for various values of s and t, representing time. The matrix elements correspond to the recurring dynamical state of a system and are given by the following equations

$$R_{s,t} = \Theta(\varepsilon - ||x_s - x_t||); s, t = 1,2,3,\ldots,N \tag{2.13}$$

Eqn. (2.13) is in a phase space consisting of N points $x_s$is a point located on the m dimensional space,$\varepsilon$is the specified threshold, is the Heaviside step function, and $||.||$indicates the L_2 norm. In the current study $\varepsilon$is fixed at 10% of the diameter of the reconstructed phase space. $R_{s,t}$is zero if the distance between the two points $x_s$and $x_t$in the phase space is greater than ε; otherwise, it is equal to 1. Black and white points are present in the recurrent points corresponding to the ones and zeros present in the recurrent matrix. For all RPs, the main diagonal is a black line. Various characteristics of the RP can be found in Marwan[101]. The RP corresponding to a periodic system is characterized by equally spaced lines parallel to the main diagonal, unequally spaced lines parallel to the main diagonal designate quasi-periodic dynamics. For chaotic systems, the RP would consist of short, broken diagonal lines parallel to the main diagonal along with single isolated points. RP is used in the current study as a secondary measure

of quantify chaos in addition to the LLE.

The center of mass time series obtained for both the swarms was also analyzed from a multiscale entropy (MSE) perspective [102], first introduced by Costa et al.[103] as a qualitative measure for complexity. MSE can be used to determine whether a time series arises from a highly stochastic or a highly deterministic process, or, in other terms, it indicates the orderliness of a system. Multiscale Entropy is a very useful tool for evaluating the determinism or randomness in a time-series. It also measures the structural complexity in a physical system comprising of very high degrees of freedom. The Multivariate Multiscale Entropy (MSampEn) introduced by Ahmed [102] et al. is used in this study. The MSampEn algorithm proposed by Ahmed[102] uses a separate embedding dimension and time lag for a multivariate time series taken as input and is given by the following equation.

$$M_{SE}(M, \tau, r, N) = -\ln[\frac{B^{m+1}(r)}{B^{m}(r)}] \tag{2.14}$$

M is the embedding vector, τ is the time lag vector, r is the tolerance level, N is the number of data points in the time-series,$B_i^m(r)$is the frequency of occurrence, and $B_i^{m+1}(r)$ is the multivariate frequency of occurrence. MSampEn is used to find the Multiscale Entropy of the 2D center of mass time-series obtained from the Attackers and Defenders swarm, respectively. The embedding dimension and the time lags for each dimension were determined by a computer program developed by Chen[99].

In the subsequent subsections, the simulations are analyzed with respect to the increasing N_A/N_D ratio, as presented in Table 2.2. Each case is analyzed from a dynamical systems point of view and is strongly investigated for chaos. The final state of the simulation is also explored in each case, respectively. In the last sub-section, Multiscale Entropy analysis was carried out.

2.4.1 Analysis of dynamical behavior: Timeseries, phase space, and attractor visualization plots.

Starting with the first case (case 1) in Table 2.2, it was observed that the number of Defenders greatly outnumbered the number of Attacker agents. Case 1 was trivial, as the simulation outcome could be easily guessed. Fig.2.1 revealed the snapshot entire simulation; it was observed that 4 of the 5 Attacker Agents were killed relatively early in the simulation. It was also observed that one Attacker agent and 96 Defenders agents were left until the end of the simulation. The survival of one Attacker agent until the end may seem to be a bit counterintuitive; the behavior observed could be explained due to the buildup of excessive repulsive force on the remaining Attacker agent from the 96 surviving Defender agents, as all agents in this simulation are globally coupled with the force terms implicitly controlling the range of vision.

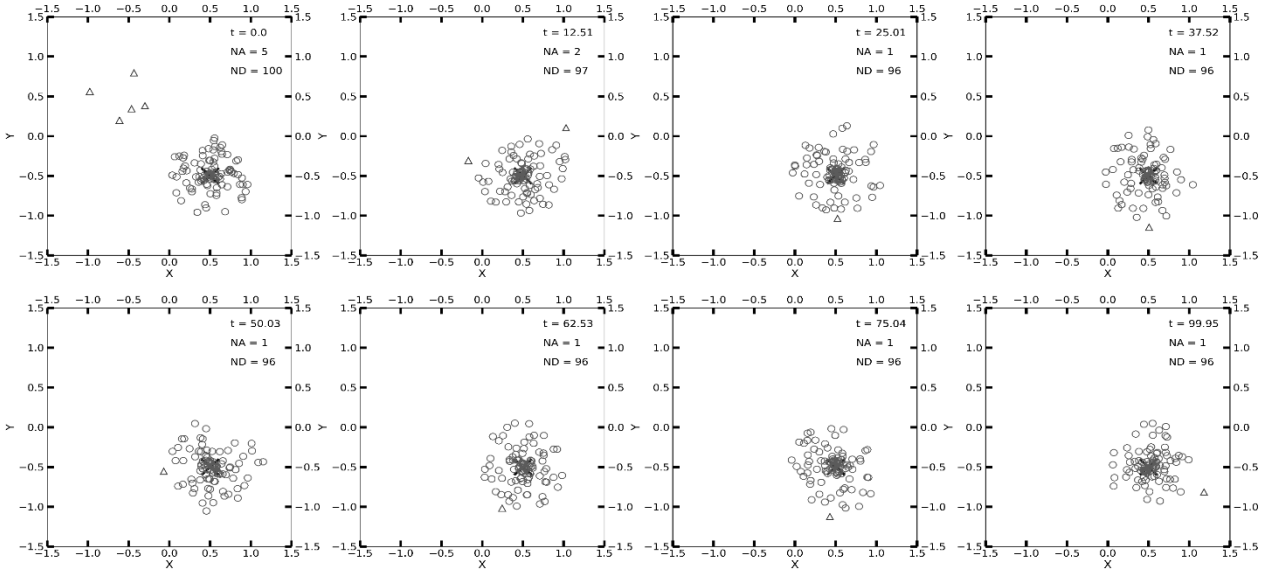

Figure 2.2: Snapshot of simulation for Case#1. Note: triangles and circles indicate Attacker and Defender agents, respectively.

The Defenders were considered dominant as they successfully protected the Goal until ssssthe end of total simulation time. The outcome was also common sense, as the Defenders greatly outnumbered the Attackers. From the time-series plots in Fig. 2.3, the agents' behavior

was highly transient with periodic and non-periodic changes in amplitude; the rapid changes in amplitude of the center indicate close interaction between Attacker and Defender agents resulting in agent compromises on either end. After the compromise of agents or agents in each swarm, the CMS was calculated only among the active agents in the next timestep, causing the center of mass to shift rapidly

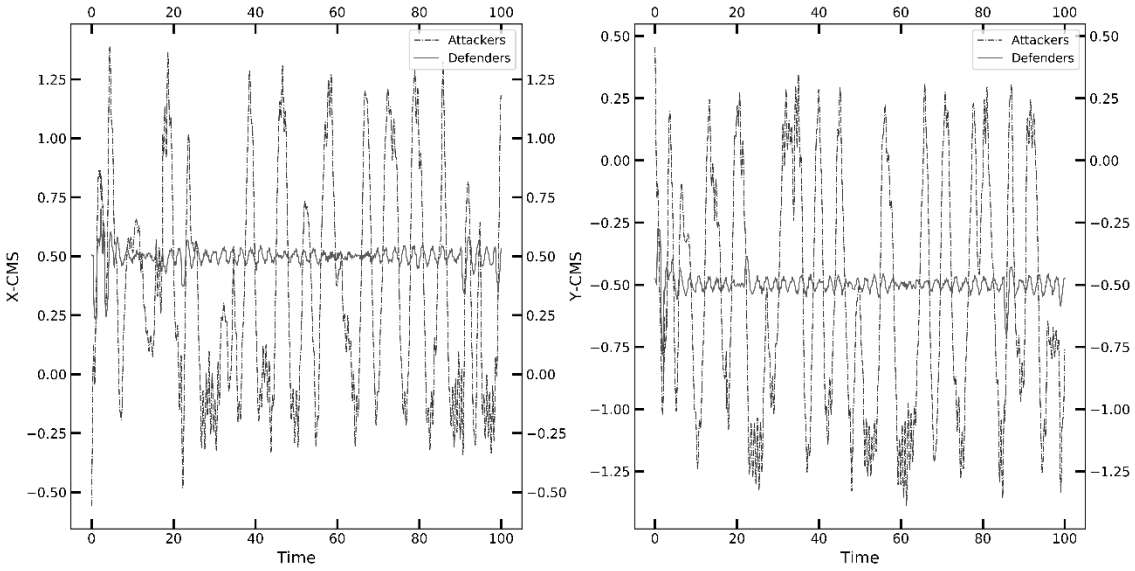

Figure 2.3: Center of mass time-series plots for Case#1

The center of mass for case#1 for both the agents of the Adversarial system was visualized in 2D space (in Fig. 2.3), revealing the existence of "transient chaos", which is defined as a situation where the trajectories leave the chaotic regime after a certain of amount of time has passed resulting in the formation of a quasi-periodic pattern of motion[104]. Figs 2.3 and 2.6 showed that the agent movements looked chaotic initially and then abruptly switched to a quasi-periodic oscillation, quasi-periodic oscillation, which lasted for the remainder of the simulation.

The initial chaotic trajectory was attributed due to the engagement between the Attacker and the Defender agents, resulting in most of the Attacker agents' death, which could be observed from Fig.2.3. As most of the Attacker agents were compromised in action, the quasi-periodic

motion of the center of the Attacker swarm is along around the Goal, suggested the few reaming Attacker agents were rotating in a quasi-periodic orbit around the Goal. The Defenders' time history indicated (Fig. 2.3) similar behavior with a lesser change in amplitude than the Attacker swarm. The Defender swarm was tightly packed due to many agents compared to the Attacker, so the swarm center's change was considerably less than its competitor.

In the end, however, as visualized in Fig.6, it was observed that the last two remaining Attacker agent moved quasi-periodically outside the sphere of influence leading to their existence until the end of the simulation. To quantize the presence of chaos, the Largest Lyapunov Exponent (LLE) was computed for the center of mass time series for both the agents using Wolf's algorithm[98] by performing phase space reconstruction. The phase space was reconstructed using the x-component of the center of mass time series to evaluate its embedding dimension by evaluating the fraction of false nearest neighbors and estimating the time-delayed mutual information time-series [99]. The LLE for both the agents converged at 0.022 and 0.024, respectively, thus proving the existence of chaos.

The chaotic behavior was also further quantified by the Recurrence plots (RP) in Figs.2.4 and 5 for the Attackers and the Defenders, respectively. The RP for the Attackers depicts a main diagonal line and short, broken lines parallel to the main diagonal line. Sporadic points were also noticed on either side of the diagonal, indicating sustained chaotic dynamics for the Attackers. The RP for the Defenders (Fig. 2.5) depicts two distinct time windows indicating "transient chaos" for the Defenders.

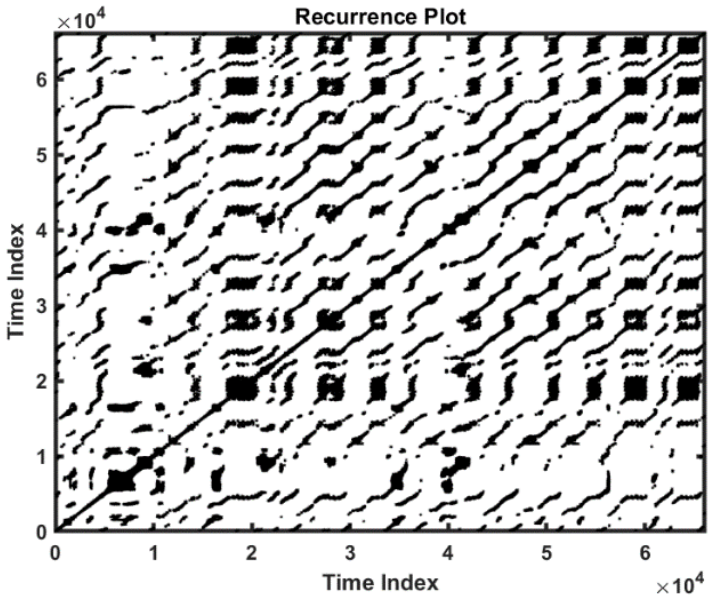

Figure 2.4: Recurrence plot of Attackers for Case#1

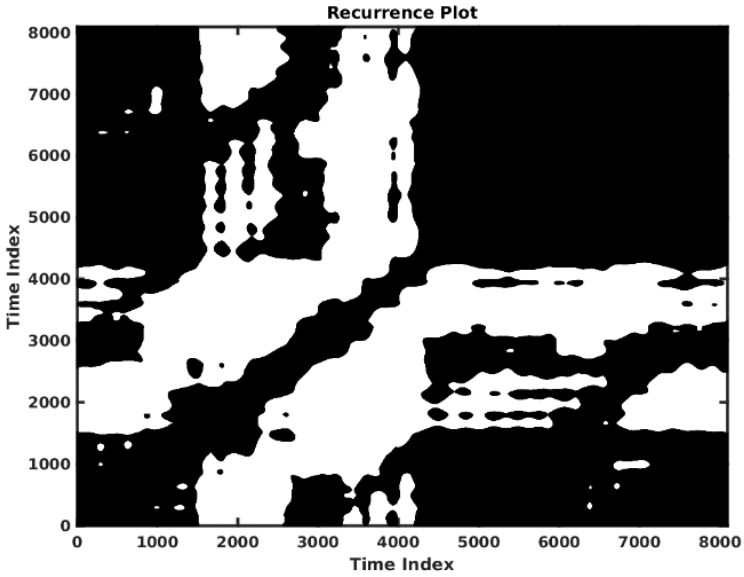

Figure 2.5: Recurrence plot of Defenders for Case#1

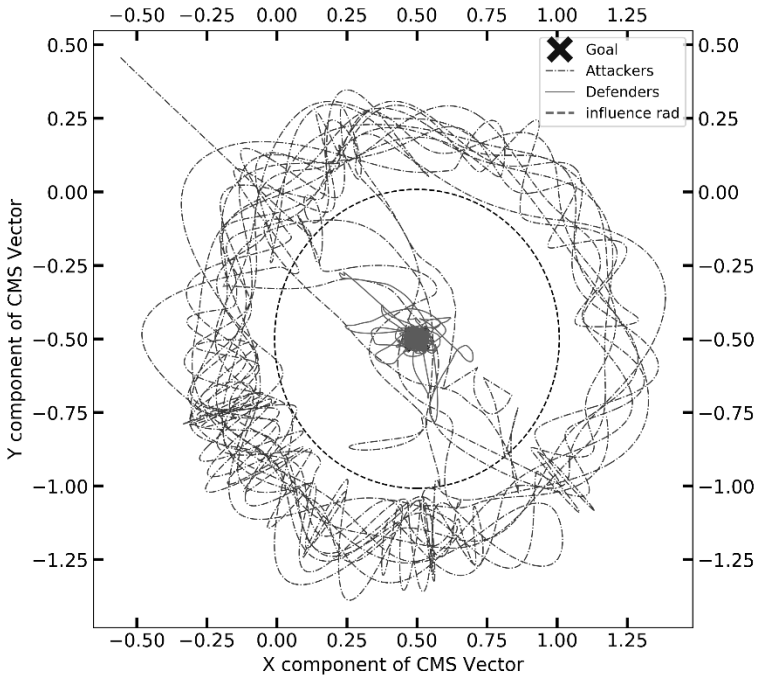

Figure 2.6: Center of mass of the Adversarial swarms in case#1 visualized in 2D space.

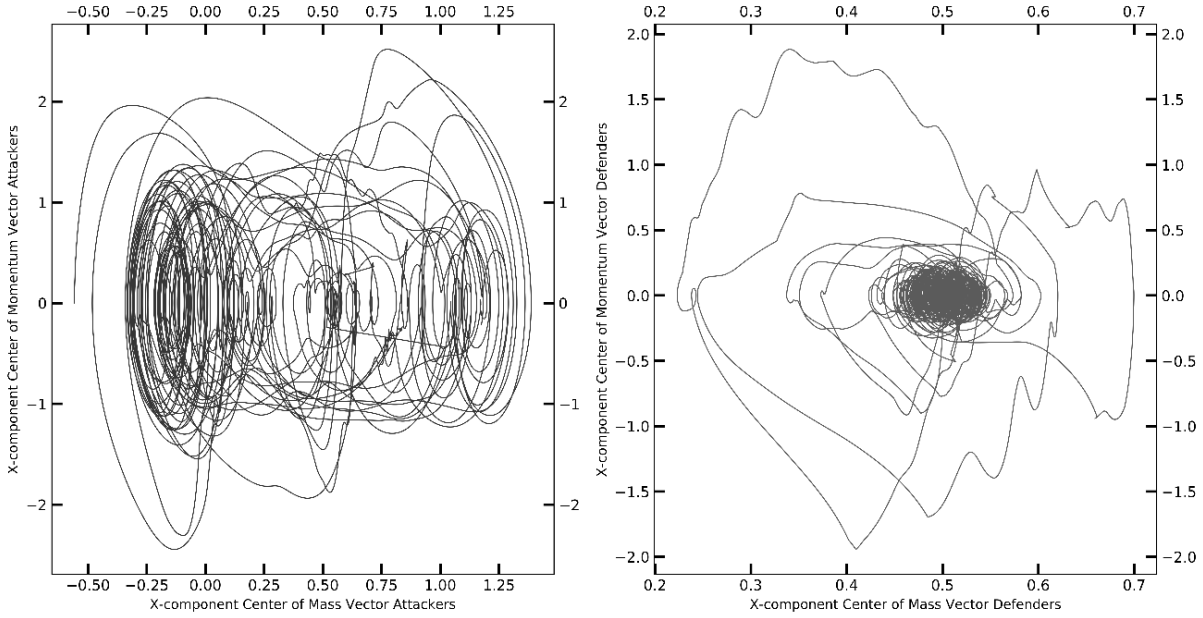

Figure 2.7: Attractor visualization for the center of mass for each swarm respectively in Case#1

From Fig. 2.7, it was observed that for the Attackers, there existed more than one dynamical attractor leading to the establishment of multiple local equilibrium points for the Attacker swarm before it settled down to a quasi-periodic orbit around the Goal. This behavior was attributed to the Attackers and Defenders' engagement, resulting in the compromise of

agents killed in action. As the compromise of most of the Attacker agents took place quickly in the initial part of the simulation, the local equilibrium points of the Attacker agents also changed rapidly; in the latter half of the simulation, however, as only one Attacker agent is left, it is almost locked in a quasi-periodic orbit around the Goal. In the Defenders' case, however, 3 local equilibrium points were observed, among which the central point was much more pronounced than the others. The outer points observed from the left plot of Fig.2.7 resulted from initial interaction between the Attacker and the Defender swarms, where some of the Defenders initially translated towards the attacker swarm due to strong attraction. The Defenders eventually traversed out of the sphere of influence and returned to the sphere of influence as they lost their attractive force once they ventured out of the sphere of influence.

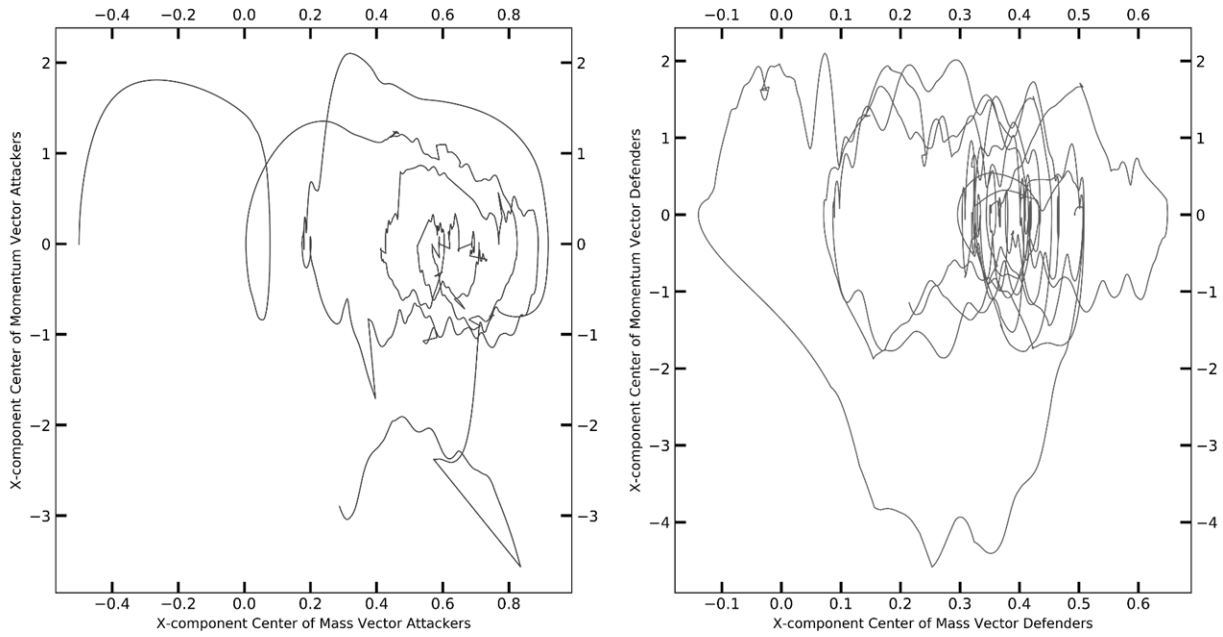

Figure 2.8: Attractor visualization for the center of mass for each swarm respectively in Case#8

Defender agents rotated quasi-periodically around the Goal with the Attacker agents' occasional attraction as it crossed the sphere of influence. The presence of multiple local equilibria for both classes of agents suggested the presence of many interdependent thresholds, which caused a rapid shift of the dynamical attractors causing rapid and drastic changes in the

system, which in the present study can be attributed to the compromise of multiple agents; as the two interacting swarms that were globally coupled intrinsically (within the same class of agents) and extrinsically (between the members of the two competing swarms). Such interdependent thresholds are often observed in the earth's climatic system, essentially a complex system, wherein abrupt changes are persistent and unpredictable [49]. Thus, the current competing swarm system under discussion was regarded as a complex system. Similar dynamical behavior was also obtained in cases 2 and 3 with multiple dynamical attractors; in these cases, the Defenders emerged as the dominant swarm.

Starting from case#4 (NA=15, ND=90) onwards, a decrease in the number of local equilibrium points for the Attackers and an increase in the same for the Defenders was observed, which was mainly due to a relative decrease in the total run time of the simulation, which meant that all the Defenders were able to intercept the Attacker agents in early on the simulation. There was a decrease in the simulation's average total run time for all the trials (as presented in Table 2.2), which will again be discussed in detail in this paper's succeeding subsection. This trend was observed in the overall dynamical behavior and continued until case 8(NA=35, ND=70). The attractor visualization plot for case#8 can be found in Fig2..8.

From case #9 onwards, as the number of Attackers considerably increased and vice versa. The Attackers' equilibrium points appeared centrally around the Goal. However, for the Defender agents, multiple equilibrium points were still observed, which was due to the high degree of interaction of the Attackers with the Defenders, which caused the system to arrive at the final state much faster than the initial cases, as outlined in Table 2.2. The Attacker agent's strong interaction was attributed to the decrease of the globally coupled repulsive force between an Attacker agent with all the Defender swarm agents, making them more prone to interact with

the Defender agents.

Fig.2.9 revealed the snapshot of the entire simulations taken at 8 equated timestamps throughout the simulation. In case 12, the interacting Defender and Attacker swarms were made up of an equal number of agents (50 each). Each swarm was made up of 50 agents; respectively, it was observed that the Attackers and the Defender swarm interacted actively from the onset of the simulation.

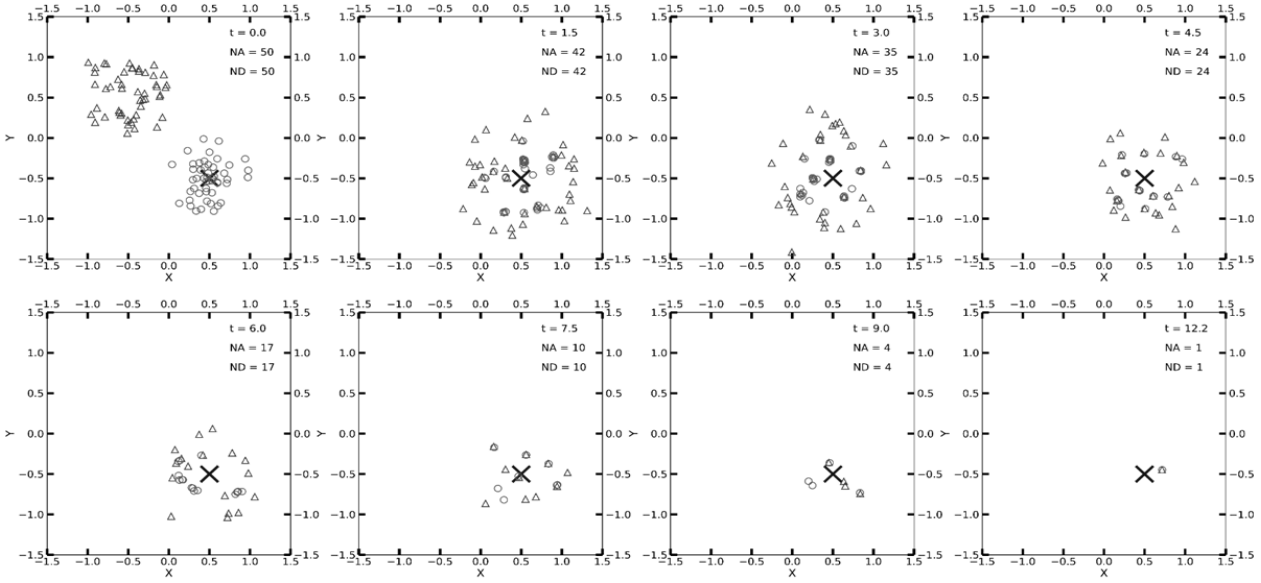

Figure 2.9: Snapshot of simulation for Case#12. Note: triangles and circles indicate Attacker and Defender agents, respectively.

During the engagement, there was a substantial loss of agents from early on. Fig. 2.9 also revealed that the loss of agents happened at almost the same proportion, indicating that agents were engaging on a one-on-one basis with the spatially closest adversarial counterpart before the collision, ultimately resulting in the pair's loss. This behavior can be found towards the end of the subplots in Fig.2.9.The Defenders dominated this simulation's final state as their ultimate objective was to protect the Goal from the Attackers. This case was also run 1000 times like the other cases. A detailed study for all the other cases revealed that the agents' interaction was not

always be binary (one-on-one) in nature. Some of the cases during the Monte Carlo analysis reveal multiple agent compromise (more than 2).

The center of mass time histories of both the swarms was plotted in Fig.2.10, revealed the presence of highly transient behavior with somewhat irregular-shaped nonsmoothed peaks and troughs. The irregularity in the peaks and troughs was due to the loss of agents in rapid progression due to close and rapid engagements. The plots revealed the existence of transient chaos like Case#1. The LLE calculated for this case was 0.003 and 0.011 for the Attacker and the Defender swarm, respectively. Near the end of the simulation, it was observed that both the agents' trajectories converged to a quasi-periodic orbit around the Goal. This behavior was observed from the phase space plots in Fig.2.11. Initially, strong interaction between the Attacker and the Defender agents resulted in rapid loss of agents, causing irregularly shaped orbits around the Goal. The center of mass shifted rapidly due to the loss of many compromised agents.

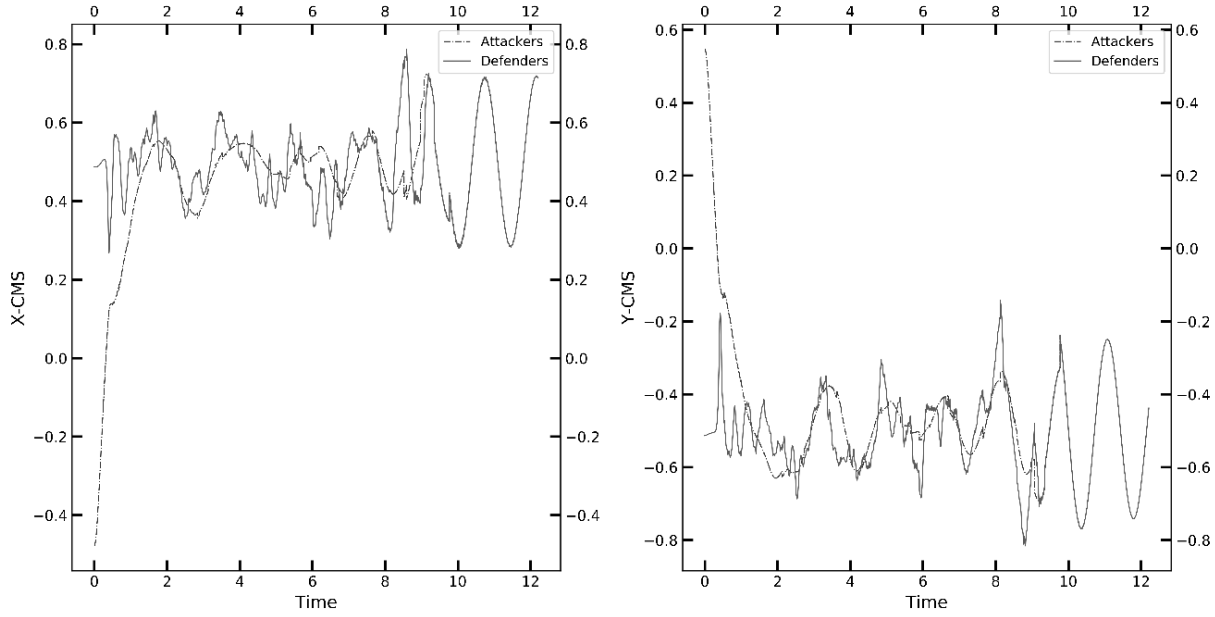

Figure 2.10: Center of mass time-series plots for Case#11

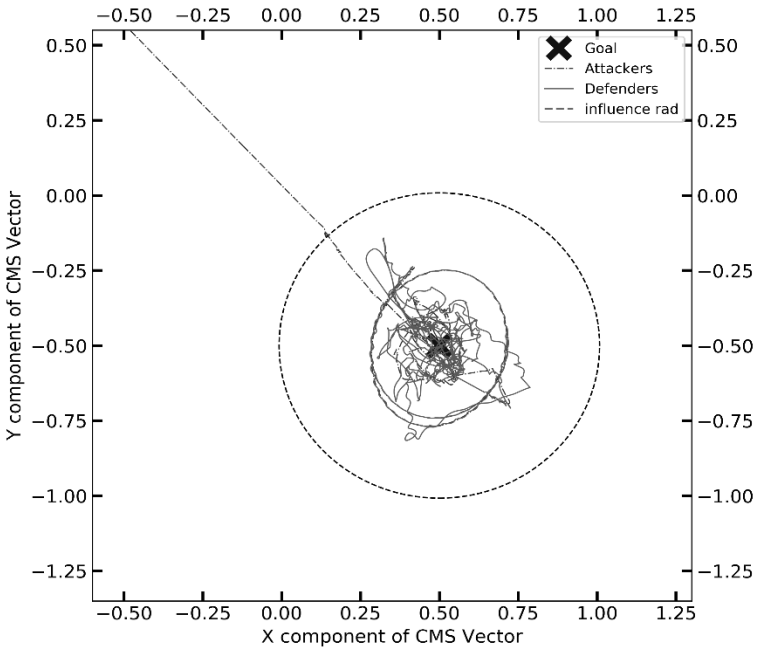

Figure 2.11: The simulation described in Case#12 is visualized in 2D space.

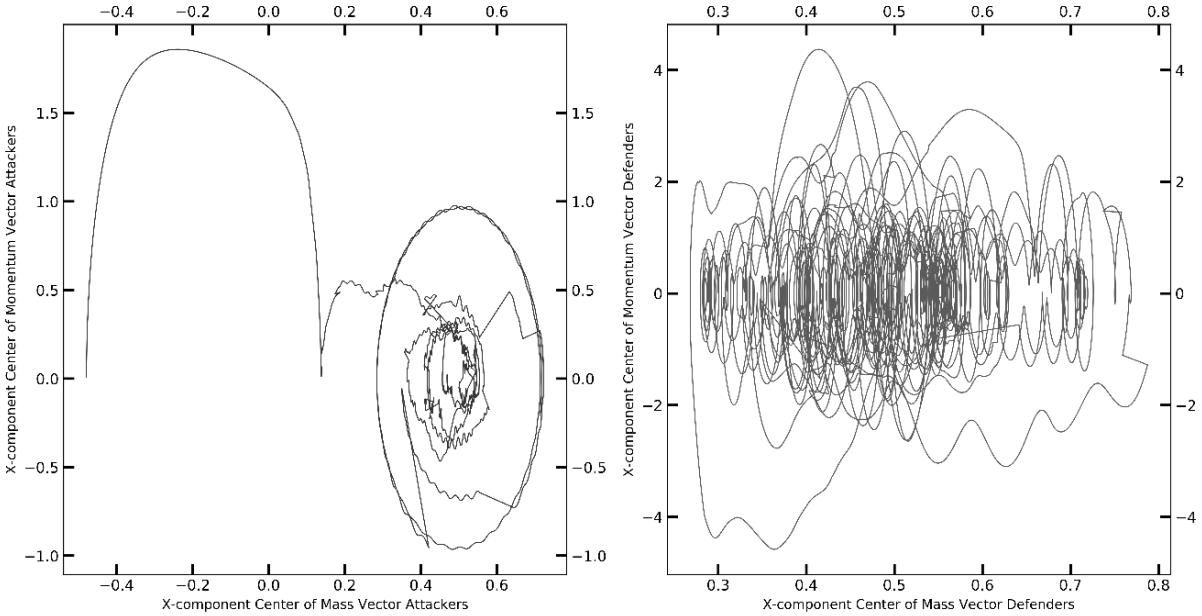

Figure 2.12:Center of swarm attractor visualization for Case#11

Fig. 2.12 revealed a single equilibrium point for the Attacker swarm around the Goal and multiple local equilibrium points for the Defender swarm. This pattern was caused by the Goal being surrounded by the Attacker agents while constantly rotating around it. The Defender

swarm center constantly shifted towards the closest Attacker agents, leading to the formation of multiple local equilibrium points.

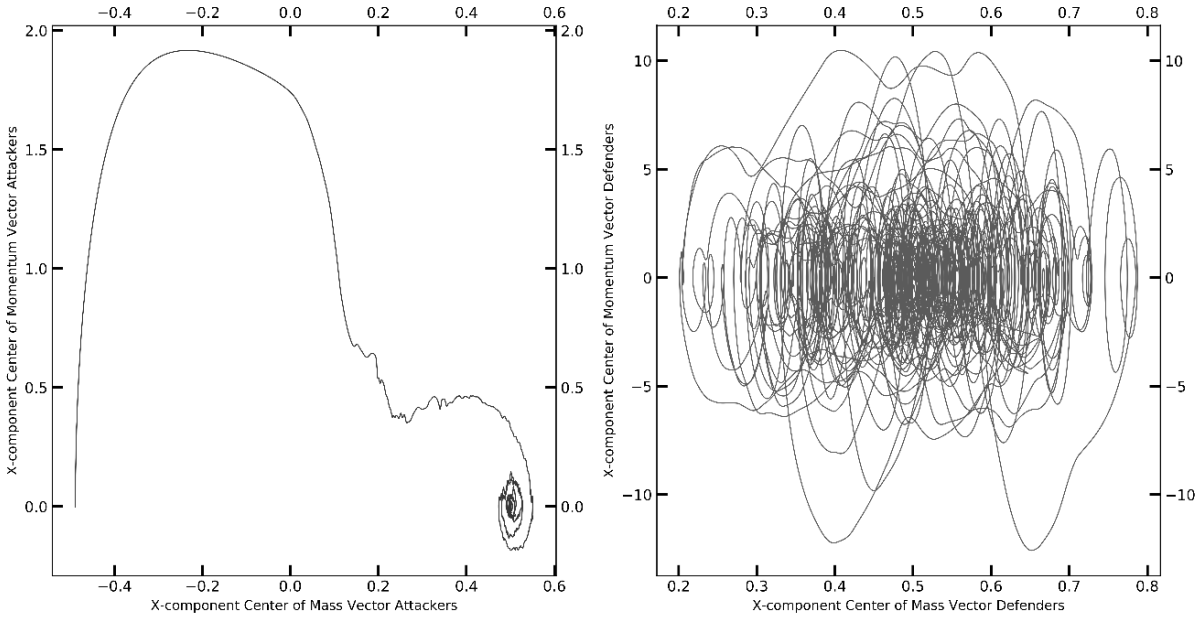

Figure 2.13:Center of swarm attractor visualization for Case#18 (NA=85, ND=15).

A similar pattern of dynamical behavior continued with reducing Defender agents and the increase of Attacker agents beyond case#12. The rotation of the Attacker agents around the Goal was also significantly reduced as the final simulation state is attained faster. The attractor visualization plot for case 18 (NA=85, ND=15) can be visualized in Fig.2.13, revealing the shrinkage of the basin of attraction for the Attackers.

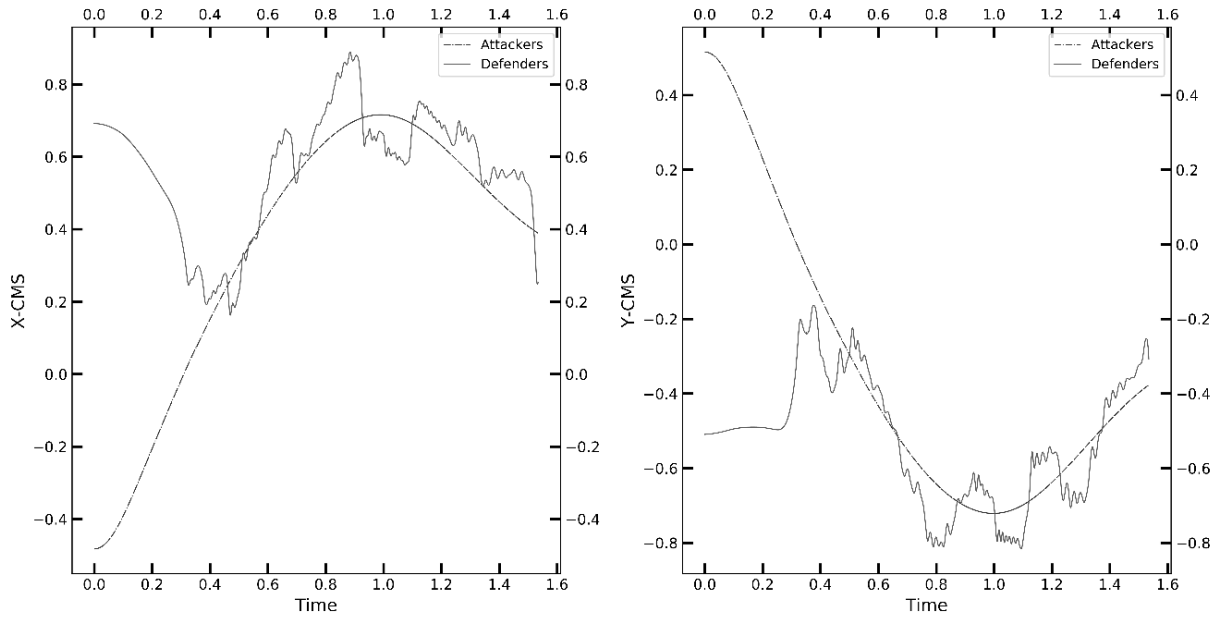

Figure 2.14: Center of mass time-series plots for Case#21.

Case#22 was the last case, essentially the converse of case#1; the Attacker agents vastly outnumbered the Defenders. The final state was trivial; it could be observed that all the Defenders agents were compromised quite early in the simulation. This behavior was expected as the Attacker agents experienced relatively less repulsive force from 5 Defender agents. The Attackers initially approached the Defenders with an almost linear trajectory followed by a minor rotation around the group (see Fig.2.15). The Defender swarm underwent erratic motion around the Goal because it tried to engage with the Attacker agents closest to the Goal, causing the center of mass to change rapidly. The close engagement with an outnumbered Defender swarm caused the rapid compromise of agents on both sides as the Defender agents could easily intercept Attacker agents, which were in a very close formation around the Goal inside the sphere of influence. The time histories of the center of mass in Fig.2.14 revealed steep peaks and troughs for the Defenders, which was caused due to the rapid compromise of agents in the outnumbered Defender swarm, causing the center of swam to shift rapidly. The Attacker swarms comprising a much greater number of agents do not show steep peaks or troughs as the center of the swarm

does not appreciably change due to their close formation on the Goal and the death of relatively few agents compared to the entire population of agents.

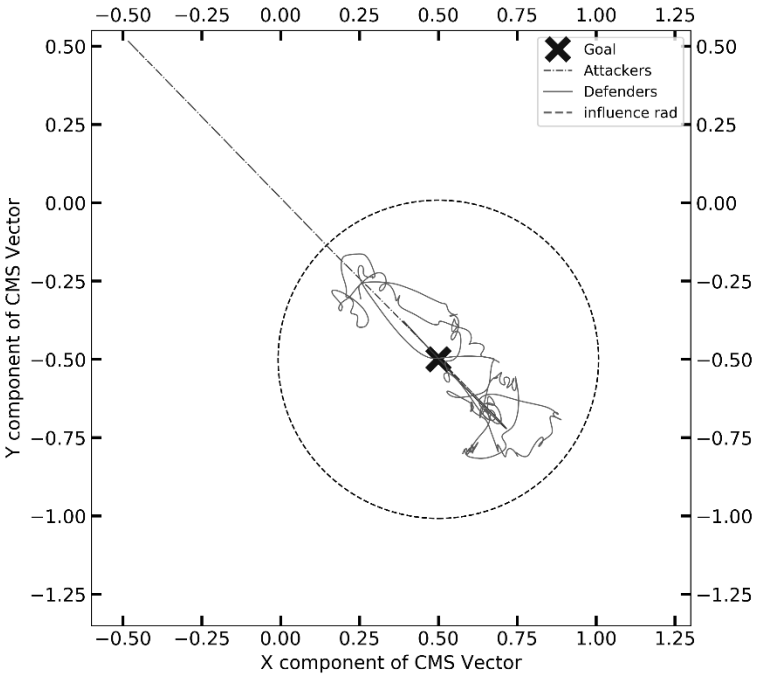

Figure 2.15: The simulation described in Case#21 is visualized in 2D space.

2.4.2 Final Simulation State

The system's final state, which is the outcome of the two swarms' interaction, was of immense importance as it determined which class of agent dominated the simulation in the end. A flag classified the dominant swarm in the simulation, defined as +1 if the Defenders emerged dominant and -1 for the Attackers. As discussed in this paper's introductory sections, the Defenders emerge dominant if the Goal was protected from the Attackers until the end of the maximum total time allotted for the simulation or if the Attacker swarm was compromised before the total time. On the other hand, the Attackers emerged dominant due to the compromise of the Defender swarm or the Goal's breach. The final state was found out by defining a Dominant Simulation State metric is calculated for each case presented in Table 2.2. The simulation Dominant Simulation State metric is defined as the simple product of flag, the average time of

run for all the 1000 trials in the Monte Carlo analysis, and the number of times a particular swarm emerged dominant(wins). The Dominant Simulation State metric is plotted against the agents' ratio on a semi-log scale (x-axis only), which can be found in Fig 2.17. The random initial condition and the high degree of freedom of the system caused the nature of the swarm agents' interaction to be stochastic in nature, making the ending of each simulation unique.

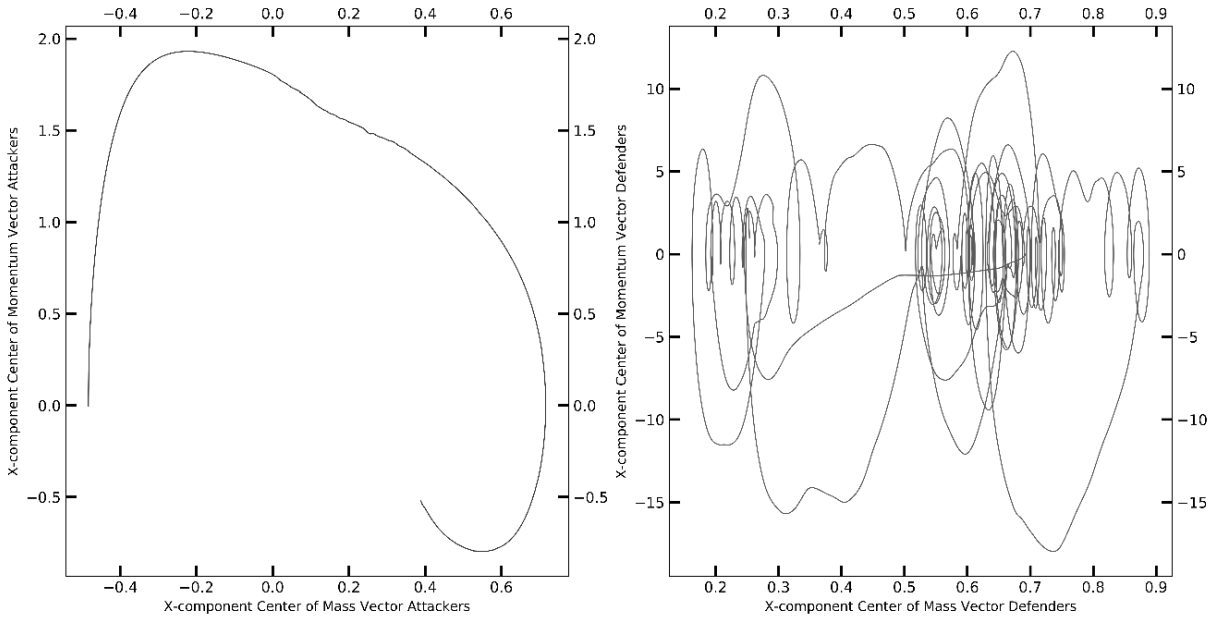

Figure 2.16: Center of swarm attractor visualization for Case#21.

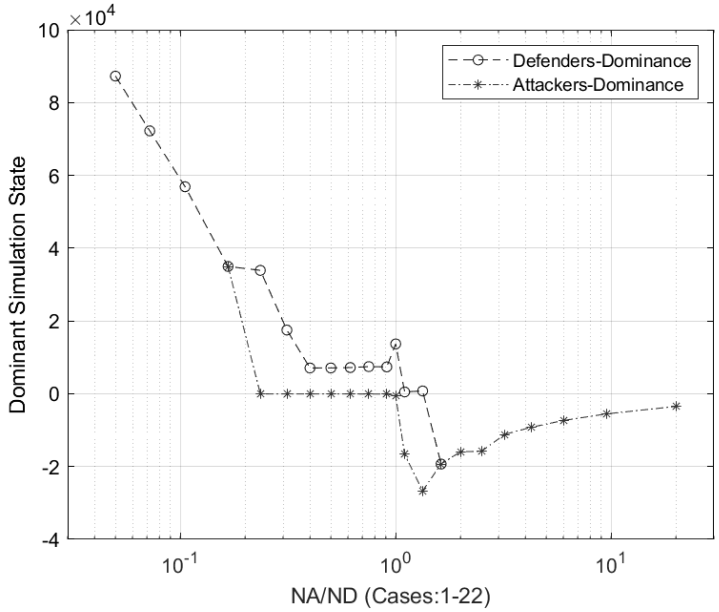

Figure 2.17: Plot of Simulation Winner Outcome versus Population of Attacker and Defenders (NA/ND). The x-axis is plotted in a log scale for the clarity of the figure.

Initially, the Defender swarm exclusively emerged dominant for cases#1-4, which indicated that the Defender swarm could protect the Goal by intercepting Attacker agents or by preventing them from reaching the Goal. It was also observed from Fig.2.18 that most of the Defender dominance was due to the Goal protection, which meant the Defenders were successfully able to protect the Goal until the end of maximum permissible simulation time. This behavior was expected due to the excessive repulsive force on relatively few Attacker agents compared to the significant number of Defenders. In relatively few cases, the Defender swarm was successfully able to intercept and compromise the Attacker swarm as its movement was highly constrained around the Goal. It was observed that starting from case#5 through case#12, there were binary final states of the system, which means either swarm assert dominance. It was also noted that the Defenders emerged as the dominant party in most of these simulations. The dominance was profoundly due to the compromise of the Attacker swarm due to the strong interaction. In relatively few cases, the Defenders emerged victorious by standing ground until the end. This trend continued until case#12. Much of the Attacker swarm's marginal winning until case#12 (NA/ND=1) was caused by the breach of the Goal, while the minority of the wins was due to the total compromise of the Defender swarm. Beyond case#12, as the number of agents in the Attacker swarm increased considerably, the Attacker swarm's dominance increased mostly by compromising the Defender swarm compared to the Goal's breach, which continued for the remaining cases in Table 2.2. However, the Defender swarm only exerted marginal dominance from case#12 onward until case#14 by preventing the breach of the Goal.

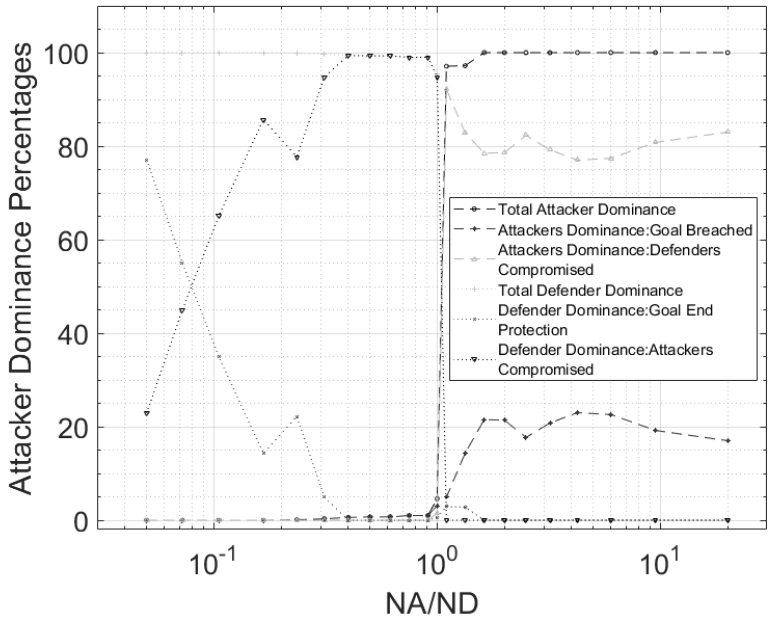

Figure. 2.18: Plot of Simulation Winner Outcome versus Population of Attacker and Defenders (NA/ND). The x-axis is plotted in log scale for clarity

2.4.3 Largest Lyapunov exponent for all the cases presented in Table 2

The Largest Lyapunov Exponent (LLE) for the center of mass time series was calculated for the Attacker and Defender agents using Wolf's algorithm[98] as outlined in Eqn. (2.12) for all the 1000 trials as outlined in Table 2.2.

The average, median, maximum, and minimum LLEs found out from the trials were plotted in Fig. 19 and 20 for the Attacker and Defender swarm, respectively. It was observed that average LLE hovered in and out of zero for both the agents, indicating that the system was on the Edge of Chaos [83-85].

A dynamical system crosses the boundary between a highly deterministic system and a chaotic one in this region. It is believed that at the "Edge of chaos," the complexity of a dynamical system increases and it has the greatest computational capacity[83]. From Fig. 2.17 and 2.18, it appeared that the average LLE for the Defenders was greater than zero for all the cases presented in Table 2.2 and the average LLE for the Attackers is less than zero beyond case 9. The minimum

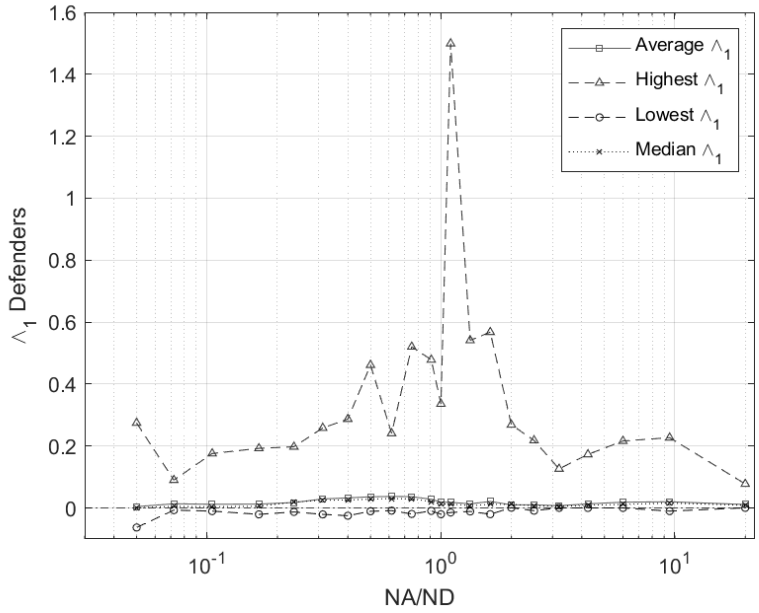

Figure. 2.19: Largest Lyapunov Exponent (LLE) of the center of mass time series of Defender swarm versus the Ratio of the Number of agents initially making up the Attacker and Defender swarms, respectively, for 1000 trials.

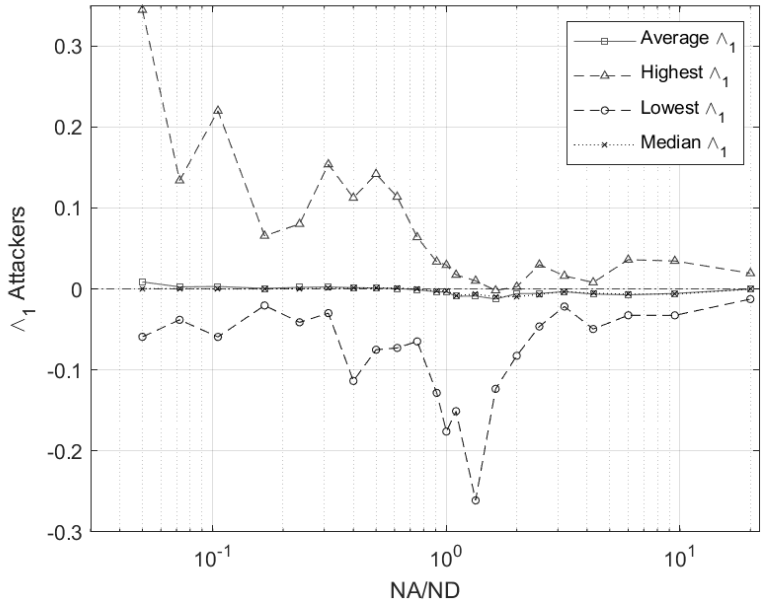

Figure 2.20: Largest Lyapunov Exponent (LLE) of the center of mass time series of Attacker swarm versus the Ratio of the Number of agents initially making up the Attacker and the Defender swarm, respectively for 1000 trials.

and the maximum LLE for all the cases are positive and negative, respectively, indicating the presence of chaotic and non-chaotic solutions. The interacting swarm system, in this case, could

be concluded to be in the 'Edge of Chaos,' which caused a rapid change in the dynamics of the system. From Fig.2.17, it was evident that from cases #5 through 7, even though the number of Defender agents was large compared to the number of agents in the Attackers, it was tempting to conclude that the outcome of such cases would be trivial (Defenders exert dominance). On the contrary, a marginal number of cases where the Attacker swarm can successfully breach the Goal. The system hovers at the 'Edge of chaos,' causing rapid change in the system's underlying dynamics; thus, it may be concluded that some solutions appear to be chaotic, ultimately affecting the system's final state.

Figs. 2.19 and 2.20 also reveal that the highest LLE reorder for the entire study occurs at case 13 (NA=55, ND=50), equal to 1.55 was for the Defenders. The RP was also found for this case indicated the presence of chaotic behavior as indicated by the broken line parallel to the main diagonal line and sporadic points in Fig 2.21. LLEs was obtained for cases 11 through 13 was notable, as the number of Attackers and Defender agents making up their respective swarms was equal or numerically very close to each other. A histogram was plotted for the LLEs calculate for both agents in case#12 was plotted in Fig 2.17. It was observed that both the swarm have a prominent central peak at -0.006 and 0.010 for the Attackers and Defenders, respectively, reaffirming both the swarms are zoning around the edge of chaos. The histogram further indicated that most of the Defender swarms for the trials in case#12 had a positive LLE. The highly chaotic behavior was due to the Defender swarm's rapid engagement with the many Attacker agents in multiple directions. As a consequence of rapid engagement, multiple agents on either side were compromised. Most of the Attacker agents, on the other hand, have negative LLE as they mainly exhibit periodic or semi-periodic movement around the Goal while moving in and out of the sphere of influence while interacting with the Defenders.

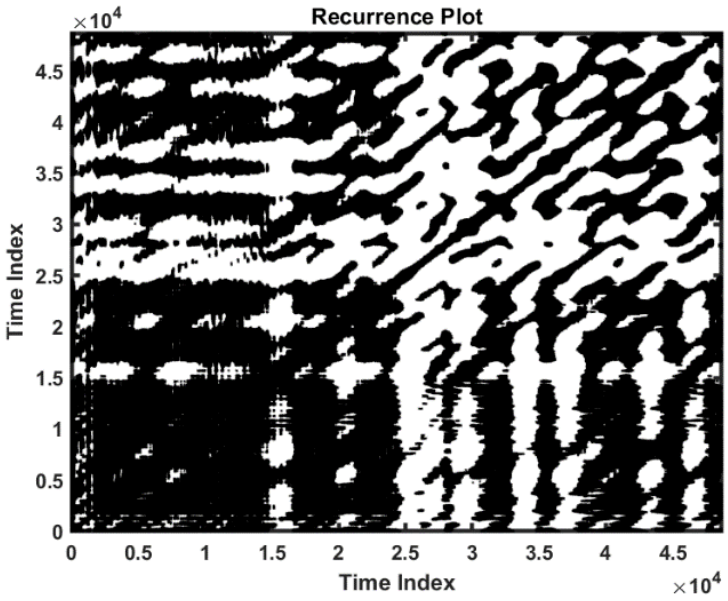

Figure 2.21: Recurrence plot for the case where max LLE obtained for Defenders in Fig.19

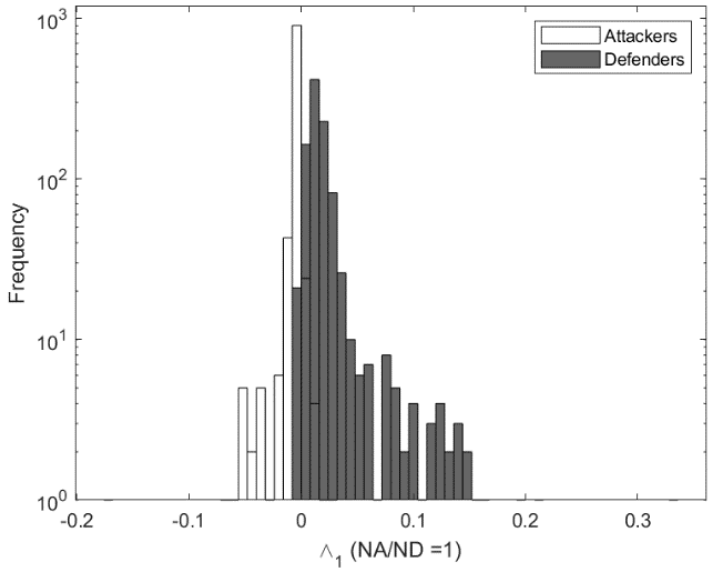

Figure 2.22: Histogram of LLEs of the center of mass time series for case#12(NA=50, ND=50) for 1000 trials.

2.3.4 Multiscale Entropy analysis

Multiscale Entropy was used to quantize the degree of randomness of the Defender and the Attacker swarm, respectively. Multiscale Entropy was calculated for all the cases in Table 2 for time scales from 1 to 20, which revealed that it is monotonically increasing. A concise picture of the change in MSE for all the simulation cases studied in Table 2.2 was presented by plotting

the ratio of the Multiscale Entropy calculated at scales 20 and,1 respectively, and can be found in Fig.2.22.

The MSE of the Defender swarm was greater than the Attacker swarm for all time scale indicates that the Defenders were dynamically more complex than the Attackers. The ratio of the MSE for the Defenders for all the scales indicated monotonically increasing behavior. The Attackers also exhibited the same trend except until case 16, beyond which the ratio of the MSE is ~1.01, indicating that the MSE increase was not appreciable across scale 1 to 20. A detailed study of these cases revealed that the MSE did not show an appreciable increase when plotted from scales 1 through 20. Thus, overall, it can be concluded that the interacting swarm system exhibits the behavior of a complex system.

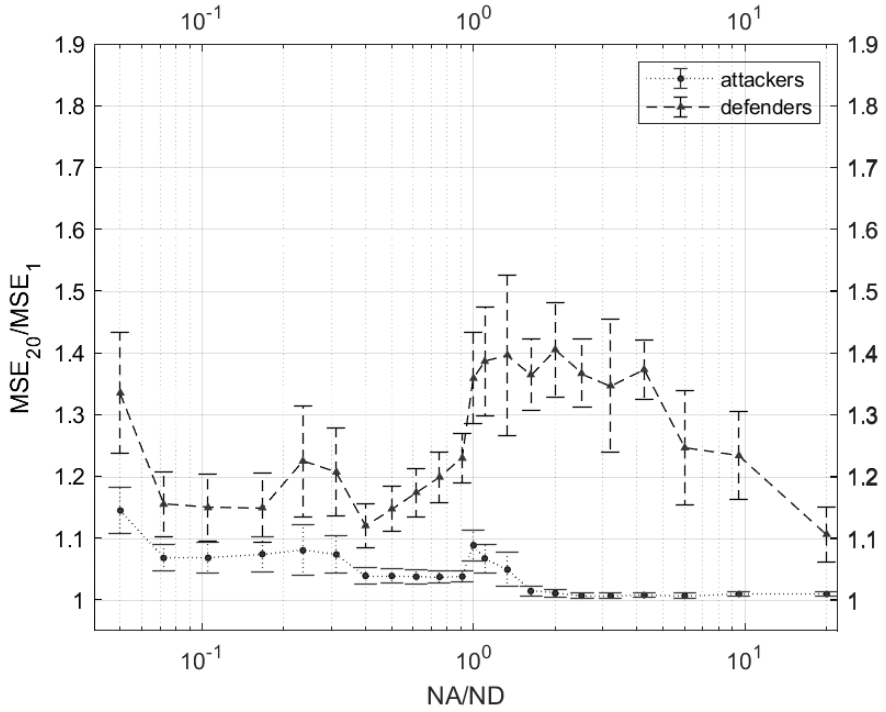

Figure 2.23: Ratio of Multiscale Entropy calculated at scale 20 and scale for the center of mass time series for Attacker and Defender swarm respectively, for all the cases presented in Table 2. The error bars represent the sample standard deviation in each case.

This overall trend can be explained from a philosophical standpoint that a complex system is often defined as more than just a regular combination of its constituents. A standalone

system constituent cannot explain the complex emergent behavior observed by the system overall. In the present problem, the Attackers and Defenders swarm are made of simple agents that exhibit highly emergent behavior in swarming while interacting with each other in an adversarial manner. The MSE was also used as a measure of the 'orderliness' of a system; the MSE indicated that the Defenders were less ordered as a system when compared to the Attackers; this was expected because when the Defender swarm engages with the Attacker swarm the Defenders are forced to be constrained in a relatively smaller perimeter defined by the influence radius used in the Attractive Morse force between the Defender agents and the Goal. The phenomenon was also evident from multiple attractor visualization plots presented in the previous sections. The Defenders' constrained motion was dynamically more complex than the unconstrained motion of the Attackers in 2D Euclidian space. This phenomenon was backed by the dynamical behavior observed in Figs. 2.3 and 2.7, respectively.

It can also be concluded from Fig. 2.23 that the Defenders have higher mean multiscale Entropies occur from cases #11 through 16; these cases also had an extensive variation as indicated by the standard deviation error bars. On the other hand, the attackers exhibited relatively higher mean MSEs from 1 through 6 and then again from 12 through 14. Cases 11 through 14 exhibited overall the most dynamically complex behavior for both agents; the number of Agents in both the swarms were either equal or numerically very close. These cases also exhibited binary final states, as indicated in Fig. 2.17, along with the maximum variation of the LLEs is observed from Fig. 2.19 and 2.20. In the initial few cases (cases 1 through 5), the Attacker swarm exhibited relatively higher MSEs as the size of the Attacker was considerably smaller compared to the Defender swarm, thus causing greater randomness in the overall interaction when studied over 1000 trials. Beyond case #15, the Attacker swarm emerged as the

dominant swarm in all trials because of being heavily outnumbered by the Defender swarm. Fig. 23 revealed a drop in the randomness of the simulation for all cased beyond case#15. The Attacker swarm quickly exerted it dominance which was expected as the Defender phase plots contained erratic trajectories around the Goal as it was overwhelmed by a vast number of Attacker agent engaging from all sides, all these factors caused a drop in the MSE of the Attacker swarm along with its overall variation in all the trials; as for the Defenders complex dynamics also existed for all the cases which lead to a higher degree of random behavior causing the MSE to be relatively higher than the Attackers.

The MSE calculation of the Attacker and Defender Swarm overall indicated the presence of complex dynamics which can be used to rule that the current scenario considered for the simulation can be referred to as a complex system; the MSE entropy plots for the Attackers and the Defenders also exhibits an intermediate level of randomness which lies between a perfectly deterministic system and a completely random system. Thus, it can be concluded that the semi-hybrid approach assumed at the beginning of this paper is successful as the simulation can be used for meaningful engineering applications

2.4 Conclusion

The adversarial swarm model discussed in this paper comprises two types of agents: The Attackers and the Defenders, which are interacting in unconstrained 2D Euclidean space. The force terms present in Newton's second law-based governing equation can be broadly classified into two types: 'intra' and 'inter' respectively, responsible for swarming in the respective swarms and their interaction with their competing counterpart.

The Defender's main aim was to protect a point of interest in 2D space referred to at the 'Goal.' In contrast, the Attacker's main objective was to intercept the Goal while continually engaging with the Defenders. Semi-Hybrid approach was assumed in the simulations as agent

and goal compromise criteria was modelled as per real world scenarios that may arise in engineering applications. A total of 22 simulation cases were studied with a decreasing number of Defender agents and an increasing number of Attacker agents, as presented in Table 2. A Monte Carlo analysis was done by running each case 1000 times to statistically study the final state of the simulation, LLE, and the MSE. The following are the main conclusions can be drawn from the paper:

- The overall model discussed was a generic dynamical system-based Agent based Adversarial Swarm model that was solved using a rigorous numerical procedure. This model is generic enough and can be adapted to a wide range of engineering applications.
- In the spectrum of the study conducted, the initial cases (#1-#4) revealed transient chaotic behavior with multiple local equilibrium points for both parties that reaffirmed the existence of complex dynamics.
- Beyond case #10, the number of local equilibrium points reduces for the Attackers as it converges to a central equilibrium point. On the contrary, multiple equilibrium points existed for the Defender agents for all the remaining cases.
- The LLEs found out for the Attackers and Defenders from the center of mass time series for all the trials in each case, respectively, revealed the existence of both chaotic and non-chaotic solutions. The presence of chaotic behavior was also backed up the Recurrence plots. The average LLEs indicated that swarms were both on the 'Edge of chaos,' further strengthening the presence of complex dynamical behavior.
- Finally, Multiscale Entropy (MSE) was evaluated for the center of mass time-series for the swarm from scales 1 to 20. MSE, for both Attackers and Defenders, revealed the MSE the existence of complex dynamics. The MSE revealed an intermediate level of randomness for

the simulation overall thus proving the effectiveness of the Semi-Hybrid approach assumed in the Adversarial swarm model.

In the current work, the simulations considered the interaction between the Attacker and Defender swarm in an ideal environment where the interactions occur; noise effects were not explicitly modelled but implicitly considered by using random initial conditions. The work also assumed ideal and instantaneous inter and intra swarm communication and did not consider the effects of delay in communication. The position of the Goal was also assumed to be fixed for all the cases studied. The Attackers and Defenders were also globally coupled without explicit cut-offs. The forces obtained from the physics based potential functions implicitly imposed the limitation of the range of sensing or vision for the agent. Delay in communication between the agents was not considered, which is sometimes observed in the real-world case due to various external factors. Some of these limitations will serve as the basis of future works.

REFERENCES

1. Selvi, V., and Umarani, R. "Comparative analysis of ant colony and particle swarm optimization techniques," *International Journal of Computer Applications* Vol. 5, No. 4, 2010, pp. 1-6.

2. Carere, C., Montanino, S., Moreschini, F., Zoratto, F., Chiarotti, F., Santucci, D., and Alleva, E. "Aerial flocking patterns of wintering starlings, Sturnus vulgaris, under different predation risk," *Animal Behaviour* Vol. 77, No. 1, 2009, pp. 101-107.

3. Calovi, D. S., Lopez, U., Ngo, S., Sire, C., Chaté, H., and Theraulaz, G. "Swarming, schooling, milling: phase diagram of a data-driven fish school model," *New journal of Physics* Vol. 16, No. 1, 2014, p. 015026.

4. Moussaid, M., Garnier, S., Theraulaz, G., and Helbing, D. "Collective information processing and pattern formation in swarms, flocks, and crowds," *Topics in Cognitive Science* Vol. 1, No. 3, 2009, pp. 469-497.

5. Viscido, S. V., Miller, M., and Wethey, D. S. "The Response of a Selfish Herd to an Attack from Outside the Group Perimeter," *Journal of Theoretical Biology* Vol. 208, No. 3, 2001, pp. 315-328.

6. Escobedo, R., Muro, C., Spector, L., and Coppinger, R. "Group size, individual role differentiation and effectiveness of cooperation in a homogeneous group of hunters," *Journal of the Royal Society Interface* Vol. 11, No. 95, 2014, p. 20140204.

7. Bürkle, A., Segor, F., and Kollmann, M. "Towards autonomous micro uav swarms," *Journal of intelligent & robotic systems* Vol. 61, No. 1-4, 2011, pp. 339-353.

8. Al-Rifaie, M. M., Aber, A., and Raisys, R. "Swarming robots and possible medical applications," 2011.

9. Perea, L., Gómez, G., and Elosegui, P. "Extension of the Cucker-Smale control law to space flight formations," *Journal of guidance, control, and dynamics* Vol. 32, No. 2, 2009, pp. 527-537.

10. Reynolds, C. W. "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH computer graphics*. Vol. 21, ACM, 1987, pp. 25-34.

11. Reynolds, C. W. "Steering behaviors for autonomous characters," *Game developers conference*. Vol. 1999, Citeseer, 1999, pp. 763-782.

12. Huth, A., and Wissel, C. "The simulation of fish schools in comparison with experimental data," *Ecological modelling* Vol. 75, 1994, pp. 135-146.

13. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. "Novel type of phase transition in a system of self-driven particles," *Physical review letters* Vol. 75, No. 6,

1995, p. 1226.

14. Janosov, M., Virágh, C., Vásárhelyi, G., and Vicsek, T. "Group chasing tactics: how to catch a faster prey," *New Journal of Physics* Vol. 19, No. 5, 2017, p. 053003.

15. D'Orsogna, M. R., Chuang, Y.-L., Bertozzi, A. L., and Chayes, L. S. "Self-propelled particles with soft-core interactions: patterns, stability, and collapse," *Physical review letters* Vol. 96, No. 10, 2006, p. 104302.

16. Chuang, Y.-L., D'orsogna, M. R., Marthaler, D., Bertozzi, A. L., and Chayes, L. S. "State transitions and the continuum limit for a 2D interacting, self-propelled particle system," *Physica D: Nonlinear Phenomena* Vol. 232, No. 1, 2007, pp. 33-47.

17. Mesa, A. D. S., Quesne, C., and Smirnov, Y. F. "Generalized Morse potential: symmetry and satellite potentials," *Journal of Physics A: Mathematical and General* Vol. 31, No. 1, 1998, p. 321.

18. Vecil, F., Lafitte, P., and Linares, J. R. "A numerical study of attraction/repulsion collective behavior models: 3D particle analyses and 1D kinetic simulations," *Physica D: Nonlinear Phenomena* Vol. 260, 2013, pp. 127-144.

19. Cucker, F., and Smale, S. "Emergent behavior in flocks," *IEEE Transactions on automatic control* Vol. 52, No. 5, 2007, pp. 852-862.

20. Cucker, F., and Dong, J.-G. "Avoiding collisions in flocks," *IEEE Transactions on Automatic Control* Vol. 55, No. 5, 2010, pp. 1238-1243.

21. Ahn, S. M., and Ha, S.-Y. "Stochastic flocking dynamics of the Cucker–Smale model with multiplicative white noises," *Journal of Mathematical Physics* Vol. 51, No. 10, 2010, p. 103301.

22. Park, J., Kim, H. J., and Ha, S.-Y. "Cucker-Smale flocking with inter-particle bonding forces," *IEEE Transactions on Automatic Control* Vol. 55, No. 11, 2010, pp. 2617-2623.

23. Ha, S.-Y., Ha, T., and Kim, J.-H. "Emergent behavior of a Cucker-Smale type particle model with nonlinear velocity couplings," *IEEE Transactions on Automatic Control* Vol. 55, No. 7, 2010, pp. 1679-1683.

24. Stanford, C. B. "The influence of chimpanzee predation on group size and anti-predator behaviour in red colobus monkeys," *Animal Behaviour* Vol. 49, No. 3, 1995, pp. 577-587.

25. Scheel, D. "Profitability, encounter rates, and prey choice of African lions," *Behavioral Ecology* Vol. 4, No. 1, 1993, pp. 90-97.

26. Vaughn, R. L., Shelton, D. E., Timm, L. L., Watson, L. A., and Würsig, B. "Dusky dolphin (Lagenorhynchus obscurus) feeding tactics and multi-species associations," *New Zealand Journal of Marine and Freshwater Research* Vol. 41, No. 4, 2007, pp. 391-400.

27. Nøttestad, L., Fernö, A., and Axelsen, B. E. "Digging in the deep: killer whales' advanced hunting tactic," *Polar Biology* Vol. 25, No. 12, 2002, pp. 939-941.

28. Angelani, L. "Collective predation and escape strategies," *Physical review letters* Vol. 109, No. 11, 2012, p. 118104.

29. Lin, Y., and Abaid, N. "Collective behavior and predation success in a predator-prey model inspired by hunting bats," *Physical Review E* Vol. 88, No. 6, 2013, p. 062724.

30. Kamimura, A., and Ohira, T. "Group chase and escape," *New Journal of Physics* Vol. 12, No. 5, 2010, p. 053013.

31. Wang, X., He, M., and Kang, Y. "A computational predator–prey model, pursuit–evasion behavior based on different range of vision," *Physica A: Statistical Mechanics and its Applications* Vol. 391, No. 3, 2012, pp. 664-672.

32. Gaertner, U. "UAV swarm tactics: an agent-based simulation and Markov process analysis." NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF OPERATIONS RESEARCH, 2013.

33. Luke, S. "Multiagent simulation and the MASON library," *George Mason University* Vol. 1, 2011.

34. Strickland, L., Day, M. A., DeMarco, K., Squires, E., and Pippin, C. "Responding to unmanned aerial swarm saturation attacks with autonomous counter-swarms," *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*. Vol. 10635, International Society for Optics and Photonics, 2018, p. 106350Y.

35. Zhdankin, V., and Sprott, J. "Simple predator-prey swarming model," *Physical Review E* Vol. 82, No. 5, 2010, p. 056209.

36. Kolon, C., and Schwartz, I. B. "The Dynamics of Interacting Swarms," *arXiv preprint arXiv:1803.08817*, 2018.

37. Espitia, H. E., and Sofrony, J. I. "Path planning of mobile robots using potential fields and swarms of Brownian particles," *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 2011, pp. 123-129.

38. Rayleigh, J. "The Theory of Sound, Macmillan and co." London, 1877.

39. Gould, H., and Tobochnik, J. *An Introduction to Computer Simulation Methods: Applications to Physical Systems*: Addison-Wesley, 1996.

40. Atanassov, E. I. "A New Efficient Algorithm for Generating the Scrambled Sobol' Sequence." Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 83-90.

41. Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A. "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena* Vol. 16, No. 3, 1985, pp. 285-317.

42. Chen, Y., and Yang, H. "Multiscale recurrence analysis of long-term nonlinear and nonstationary time series," *Chaos, Solitons & Fractals* Vol. 45, No. 7, 2012, pp. 978-987.

43. Eckmann, J.-P., Kamphorst, S. O., and Ruelle, D. "Recurrence plots of dynamical systems," *World Scientific Series on Nonlinear Science Series A* Vol. 16, 1995, pp. 441-446.

44. Marwan, N., Romano, M. C., Thiel, M., and Kurths, J. "Recurrence plots for the analysis of complex systems," *Physics reports* Vol. 438, No. 5-6, 2007, pp. 237-329.

45. Ahmed, M. U., and Mandic, D. P. "Multivariate multiscale entropy: A tool for complexity analysis of multichannel data," *Physical Review E* Vol. 84, No. 6, 2011, p. 061918.

46. Costa, M., Goldberger, A. L., and Peng, C.-K. "Multiscale entropy analysis of biological signals," *Physical review E* Vol. 71, No. 2, 2005, p. 021906.

47. Grebogi, C., Ott, E., and Yorke, J. A. "Crises, sudden changes in chaotic attractors, and transient chaos," *Physica D: Nonlinear Phenomena* Vol. 7, No. 1-3, 1983, pp. 181-200.

48. Mitchell, M., Hraber, P., and Crutchfield, J. P. "Revisiting the edge of chaos: Evolving cellular automata to perform computations," *arXiv preprint adap-org/9303003*, 1993.

49. Crutchfield, J. P. "Between order and chaos," *Nature Physics* Vol. 8, No. 1, 2012, pp. 17-24.

50. Carroll, T. L. "Mutual Information and the Edge of Chaos in Reservoir Computers," *arXiv preprint arXiv:1906.03186*, 2019.

CHAPTER 3

USE OF HYBRID ECHO STATE NETWORKS ON THE PREDICTION OF GROUP DYNAMICS OF AN ADVERSARIAL SWARM SYSTEM

3.1 Abstract

A hybrid Echo State Network was used to predict the dynamics of an Adversarial Swarm System. The swarm consisted of two agent classes competing for dominance over a point of interest in two-dimensional space. The Defenders' primary task was to protect the 'Goal' while the Attackers' primary task was to intercept the Goal. The Adversarial Swarm system is simulated in a semi-hybrid framework. If the distance between the agents of the Adversarial Swarm is less than a predefined criterion, agents are considered compromised. Similarly, if the distance between Attacker agents and the Goal is less than a predefined criterion, the Attackers are deemed to have intercepted the Goal. Two configurations were used to evaluate the use of Echo State Networks for predicting system dynamics. Configuration 1 employed a single ESN, i.e., the patio-temporal data for all agents of an Adversarial Swarm model was used input. In configuration 2, two separate ESNs, in parallel, were used to predict Defender and Attacker swarm dynamics. Based on the results, it was concluded that the parallel ESN configuration was more effective in achieving qualitatively similar predictions of the dynamics for the Adversarial Swarms.

3.2 Introduction

Swarms are essentially complex systems generally characterized by non-linear dynamics; an accurate physics-based model is imperative to the holistic understanding of the underlying

dynamics. Swarms are a collection of independent, autonomous agents that are widespread in nature from ranging from ant colonies[1], flocks of birds[2], and schools of fishes[3]. Nature-inspired swarm models have been an active area of research for the past couple of decades. Simulating natural swarm behavior for engineering applications is inherently problematic as all-natural systems offer inherent flexibility and scalability, often difficult to attain using digital computers. The principle of swarming forms the basis of extensive modern-day engineering applications ranging from spacecrafts[4], UAVs[5], robots[6], and optimization algorithms[1].

Typically, two types of swarm interaction are observed: -Adversarial and Symbiotic, wherein two or more swarms either compete or cooperate over resource utilization. Symbiotic swarms found in the animal kingdom include multispecies group [26]hunting, were in different groups of species team up (cooperation) with each for hunting groups of prey. Adversarial Swarms, which are the focus of the present study, are abundant in natural environments, such as groups of predators engaging with prey groups, and can be found in aquatic and terrestrial environments. Most of such interactions take place for foraging purposes. Natural examples of Adversarial Swarms include groups of omnivorous Chimpanzees hunting groups of Red Colobus Monkeys[27] and groups of predator Lions hunting herds of Zebras[28].In the aquatic environment, a multispecies association of Dolphins with Seals and Dogfish for feeding schools of small fish[29], groups of Killer Whales, and a large number of Herring, where the former would force the latter to dive up by almost 150 meters[30], which would enable more effective foraging.

Historically, the Adversarial Swarm phenomena have been modeled as a predator-prey problem, explored by multiple researchers, including ecologists, physicists, statisticians, and mathematicians. These models can broadly be classified into three types: kinematic, lattice-

based, and dynamical models. In addition to the computational models, few experimental studies have also been conducted in the recent past. In kinematic models, the interactions between agents are typically modeled as velocity terms. Angelani [31]investigated the collective predation in a simple agent-based model capable of reproducing animal movement patterns where the individual agents were modeled based on Vicsek's self-propelled[10] agents. Lin[32] used a self-propelled particle-based model to study the predation of bats on prey. In the lattice-based models, the computation domain is divided into uniform 2D grids or lattices, which have 'states' associated with them, e.g., empty or filled. Notable lattice-based models include Kamimura et al.[33], where group chase and escape in a swarm modeled, and the study concluded the formation of highly self-organized spatial structures. Wang et al. [34]extended the predator-prey problem by adding a third species and considered the effects of stochastic vision; the study concluded a direct relationship between the predator's vision and the prey's extinction rate. Other notable works on swarms to swarm interaction includes Gaertner et al.[35], where an agent-based model based on the MASON library[36] was used to model the engagement between two groups of UAVs in 3D space. Strickland[37] studied swarm engagement during live experiments with two swarms of UAVs based on different pursuit and evasion strategies.

Dynamical swarm models are explicitly based on Newton's second law of motion, which offers accurate insights into the highly complex emergent behavior between the two swarms. Zhdankin et al. [38] studied the dynamics of a swarming predator-prey model, where each group's swarming was based on long-and short-range forces, and a non-conservative force was used to model the interaction term between the swarms. The study concluded the presence of Chaos, quasi-periodic, periodic behavior, and the existence of singularities. Kolon et al.[39] investigated the collision of two swarms made up of homogenous agents by considering the effect

of delay in communication between various agents; the study demonstrated mutual swarm capturing during the interaction, ultimately leading to the milling[18] state of motion.

Gupta et al.[105] developed a physics-based dynamical Adversarial Swarm model with well-defined intra-swarm and inter-swarm forces. The model consists of two distinct interacting swarms: The Attackers and the Defenders, which have conflicting objectives in unbounded 2D Euclidean Space. The Defenders protect the 'Goal' a point of interest in unbounded 2D Euclidian space. In contrast, the Attacker's main objective is to intercept the Goal while continually evading the Defenders. The Defenders' swarms protect the Goal by swarming around it and blocking any Attackers agent trying to reach the Goal. If an Attacker and Defender agents are very close to each other, if the distance between them is less than a predefined criterion, they were assumed to have collided and were consequently arrested for further participation. The arrested agents were inactive for the remainder of the simulation. The simulation was assumed to have a binary outcome or a final state, wherein either the Attacker or the Defenders emerged as the dominant swarm. The Attacker swarm was considered dominant if an agent in the swarm successfully intercepted the Goal during the simulation. If there were no remaining agents in the Defender swarm at any point in the simulation, then the Attackers were considered dominant. If the Defenders successfully defended the Goal before the end of the simulation or if no Attackers were left in the simulation, the Defenders were considered as the dominant swarm. If at any time during a simulation, no agents were left in either of the swarms (i,e the agents compromise each other off in the engagement), the Defenders were considered dominant in the simulation as the Goal was successfully protected from the predation of the Attacker swarm. The scenarios mentioned above formed the basis of 'Simulation Ending Criteria' presented in [105]. The non-linear time-series data obtained from the simulations performed in[105] were studied using

various tools that included time-series plots, recurrence plots, attractor plots, and the Largest Lyapunov Exponent (LLE). The system was strongly investigated for the presence of Chaos. As a vital parameter of the system, the number of Attackers and the Defender agents making up each swarm is varied to study the simulation's final state. The Largest Lyapunov Exponent for each case is also evaluated to probe the presence of Chaos. The degree of determinism in the data or the complexity of the system was found by assessing the Multiscale Entropy of the non-linear time series data.

Traditional Artificial Neural Networks (ANNs) have been traditionally limited in predicting non-linear transient times series data, which are highly non-stationary and non-cyclic in nature. ANNs have been traditionally combined with other techniques such as NARMAX[41] for predicting highly non-linear chaotic dynamics such as the Lorentz system[42], Sunspot times series[43], and downhole pressure for a gas-lift oil well[44]. Elman et al.[45] introduced the idea of Recurrent Neural Networks, which were proven to be suitable for forecasting highly non-linear time-series data obtained from a dynamic system such as the Lorentz system[46]. Training a conventional Recurrent Neural Network (RNN) is a complicated process. The backpropagation through time (BPTT) method has been partially successful in training RNNs [47, 48]. Modified RNNs such as Long Short Term Memory (LSTM) have been successful in predicting high dimensional spatiotemporal systems in the short term[49], custom architectures such as Deep Neural Network(DNN) with convolutional LSTM[50] have also been successful. Oher state-of-the-art RNN architectures include Gated Recurrent Units (GRU-D), which has also successfully predicted multivariate time-series with missing values by taking two representations of the missing patter, namely making and time interval. Random Recurrent Neural Network (rRNN) has also predicted the periodic non-linear Mackey Glass system[51]. One of the primary

limitations with conventional RNN as indicated by Demiris [48] is the non-convergence in the training process due to existence of bifurcations. Slow convergence and high computational costs of RNNs severely limit its engineering applications. Vanishing gradient problem[52] which severely limits the ability of a RNN to learn long data sequences.

An alternative to conventional gradient descent methods applied to RNNS was independently proposed by Jager and Hass[53] as the Echo State Network(ESN) and by Maas[54] as the Liquid State Machine(LSM) in which only the synaptic connections from the RNN to output neurons were trained by learning. The idea of ESNs can be traced back to Neuroscience, Dominey et al.[55] presented a learning algorithm about sequence processing in mammalian brains, e.g., speech recognition in the human brain[55, 56], was the precursor of the actual algorithm of ESNs. ESNs and LSMs were unified into a common research topic known as the 'Reservoir Computing' (RC) [57, 58]. In an ESN, the main task is to construct an RNN with randomly generated weights. The randomly constructed complex non-linear transformation of temporal data can be extracted from the output layer using simple techniques such as linear regression[59]. A great deal of art is needed to implement an ESN successfully, including the effective turning of many hyperparameters. Since the first decade of the 2000s, RC-ESN has been successfully implemented in a multitude of domains, including speech recognition[60], robot control[61, 62], forecasting financial markets[63], natural language processing[106], Oil and natural gas sector such as pressure estimation in gas-lift oil wells[68], detrending of non-stationary fractal timeseries[69] and finally dynamical systems [53] such as the Mackey Glass system[51]. Recent research also reveals that hardware-based Reservoir computers are also possible based on FPGA arrays[70] and carbon nano tubes[71]. These computers can be more effective than traditional software and may be suited for 'edge computing'[70] in contrary to the

current popular cloud computing trend.

The current state of the art reveals that ESNs have been very successful in predicting chaotic dynamic systems. ESNs in the contemporary literature have been found to use 3 configurations while predicting chaotic behaviors, namely, observer mode (non-autonomous or predictive mode), where model free prediction is achieved by utilizing limited state variables [75], the generative (or autonomous) mode in which during the prediction the output of a previous time-step is fed as the input in the Reservoir [76-78] and finally custom ensemble methods where ESNs are used in conjunction with knowledge-based models[79]. The authors of the previously mentioned works successfully verified and validated these techniques in the simple Lorentz system[76], Lorentz-96 system[80], Kuramoto-Sivashinsky (KS) system[75, 78], the Rössler system[75], and dynamics of excitable media such as the Barkley model and the Bueno-Orovio-Cherry-Fenton model [81]. ESNs have also successfully predicted large-scale dynamical systems such as Large Eddy simulation of an incompressible turbulent round jet by implementing massively large scale parallel reservoirs[89]. Hardware-based reservoir computers have also successfully predicted dynamical systems[70], such as the Mackey-Glass system. Krishnagopal[82] studied the effectiveness of a reservoir computer to separate chaotic signals and concluded that their results were better than the Wiener filter obtained from the same training data.

Several studies have also been conducted to have a holistic understanding of a Reservoir computer's inherent dynamics, which would enable its effective use while predicting dynamic systems. Carroll[83] used an RC-ESN at the 'Edge of Chaos'[84, 85] region to perform predictions and concluded that it does not necessarily improve the performance. Carroll [86]also conducted studies on the dimension of Reservoir computers and concluded the increase of

fraction dimension occurs inside the Reservoir with the increase of its dimension, which may adversely affect the performance of Reservoir Computer, Carroll[87] also conducted studies on the network structure of the RC-ESN. Zhang[88]conducted studies in the sensory phase coherence of two parallel reservoirs and concluded that short-term prediction is possible, but parallel reservoirs are limited in sensing the collective dynamics of a coupled chaotic dynamic of the entire system in the long run.

In the current work, Echo State Networks (ESN) code available from an open-source GitHub repository NeuronalX[107] and open-source python library ReservoirPy[108] was modified extensively to accommodate the prediction of the group dynamics of the interacting Adversarial Swarms by implementing two ESN configurations. The first configuration consisted of a high dimensional ESN used to predict the agent level dynamics of both the Attackers and Defenders. The second configuration comprised two parallel reservoirs, each individually assigned to the Attackers swarm and the Defenders swarm, respectively. These configurations worked in the generative (or data-driven) mode of operation. The following are the main intellectual contributions of this paper:

- Use of ESN in two configurations to predict the dynamics of an agent-based highly non-linear Adversarial Swarm System.
- The two Configs are hybrid in nature as simulation end criteria, goal beach criteria, and agent compromise criteria are forces upon the ESN.
- In the first Config, a single high dimensional hybrid ESN was used for predicting the dynamics of the Adversarial Swarm System
- In the second Config, a novel parallel hybrid ESN was used where independent ESNs were used to predict the dynamics of Adversarial Swarms.

This paper is organized as follows- section 3.3 discusses the methods and formulations, which contains the Adversarial Swarm model formulations and the formulations for various proposed ESN configurations. Section 3.4 includes the verification and validation of the ESN. Section 3.5 covers the results and discussion, followed by the conclusion in section 3.6.

3.3 Methods and Formulations

In this section, the detailed formulations of the Adversarial Swarm model can be found along with an in-depth approach for solving the same to obtain the simulated data. A precise formulation of a base ESN model is also presented, followed by the formulations of the two ESN configurations.

3.3.1 Adversarial Swarm model

A physics-based agent-based model is developed to study the dynamics of two interacting Adversarial Swarms: The Attacker Swarm and the Defender Swarm (hence, referred to as 'Attackers' and 'Defenders' respectively). The agents have conflicting objectives; the Defenders protect a point of interest in unbounded 2D Euclidean space by swarming around the Goal along a sphere of influence. In contrast, the Attackers' main task is to intercept the Goal while constantly trying to evade the Defenders. They actively chase the former in a perimeter around the Goal or a sphere of influence. The individual swarms in the swarm system are modeled based on a Lagrangian-based approach having primarily two types of forces- 'inter' and 'intra' swarm forces; the inter-forces are used to model the interaction between the agents of the Adversarial Swarms, respectively. The intra-forces are used to model the forces between members of the same swarm. Each swarm can be generalized as a collection of N agents in a 2-Dimensional space with position and velocity vectors. The governing equation describing the dynamics of the two interacting swarms- the Attackers and the Defenders are derived based on Newton's second

law of motion and are given by the following equations[23]:

$$\ddot{\vec{X}}_{A,i} = \dot{\vec{V}}_{A,i} = \frac{1}{m_{A,i}}\left(\sum_{\substack{Ai=1 \\ Ai\neq Ak}}^{N_A} -(\nabla\varphi_{A,ik}) + \sum_{Ai=1}^{N_D} -\nabla(k_{rep}r_{ij}^{-1}) - \nabla(-k_{obj}r_{iG}^{2}) + (\alpha_A - \beta_A|\vec{V}_{A,i}|^2)\vec{V}_{A,i}\right. \quad (3.1)$$

$$\dot{\vec{X}}_{A,i} = \vec{V}_{A,i} \quad (3.2)$$

$$\ddot{\vec{X}}_{D,j} = \dot{\vec{V}}_{D,j} = \frac{1}{m_{D,i}}\left(\sum_{\substack{Dj=1 \\ Dj\neq Dh}}^{N_D} -(\nabla\varphi_{D,jh}) + \sum_{Dj=1}^{N_A} -\nabla(-k_{att}r_{ji}^{-1}) - \nabla(\varphi_{jG}) + (\alpha_D - \beta_D|\vec{V}_{D,j}|^2)\vec{V}_{D,j}\right. \quad (3.3)$$

$$\dot{\vec{X}}_{D,j} = \vec{V}_{D,i} \quad (3.4)$$

Eqns. (3.1-3.4) are the principal equations for the Adversarial Swarm model that is subject to given initial conditions $\vec{V}_{A,i}(t=0), X_{A,i}(t=0), \vec{V}_{D,i}(t=0), X_{D,i}(t=0)$ of individual agents in the respective swarms are known. Eqs. (3.1)-(3.4) are numerically integrated using a customized fixed time-step 4th order Runge-Kutta explicit solver [69].

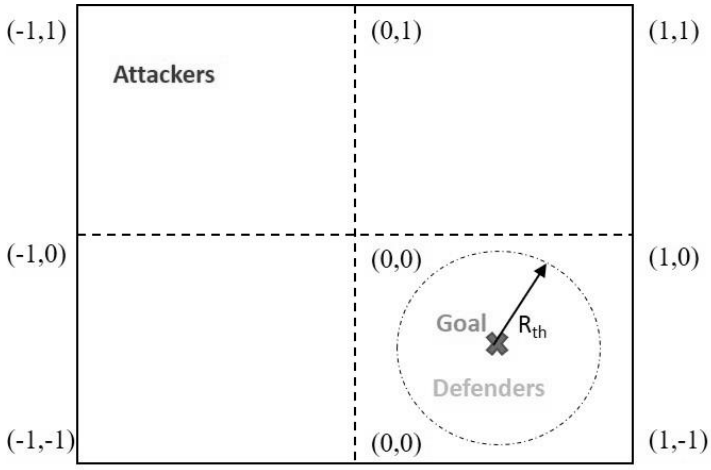

Figure 3.1: Illustration of the domain for Adversarial Swarm Simulation.

A total of 22 cases were studied for an ascending number of Attackers and descending number of Defender agents, starting with a population of 5 Attackers and 100 Defenders and ending with 100 Attackers agents and 5 Defenders agents. A Monte Carlo analysis was conducted

for initial randomized conditions to study the dynamics statistically for each case, respectively. The simulations were intensely investigated for the presence of Chaos by evaluating the Largest Lyapunov Exponent (LLE) of the center of mass time-series of each swarm, respectively, using phase space reconstruction[70] and Wolf's algorithm[71].

Table 3.1: Case Matrix for Adversarial Swarm Simulation

Case	N_A	N_D	N_A/N_D	Max Sim Time
1	5	100	0.05	100
2	7	97	0.07	100
3	10	95	0.10	100
4	15	90	0.16	100
5	20	85	0.23	100
6	25	80	0.31	100
7	30	75	0.4	100
8	35	70	0.5	100
9	40	65	0.61	100
10	45	60	0.75	100
11	50	55	0.90	100
12	50	50	1	100
13	55	50	1.1	100
14	60	45	1.33	100
15	65	40	1.62	100
16	70	35	2	100
17	75	30	2.5	100
18	80	25	3.2	100
19	85	20	4.25	100
20	90	15	6	100
21	95	10	9.5	100
22	100	5	20	100

Transient Chaos [72] was also observed for some initial cases. The overall source of Chaos in the system for all the cases studied was observed to be induced by the passively constrained chaotic motion of the Defender agents around the Goal. Multiple local equilibrium points existed for the Defenders in all the cases and some instances for the Attackers, indicating complex dynamics. LLEs for all the trials in each case revealed chaotic and non-chaotic solutions, with most Defenders exhibiting chaotic behavior.

Overall, the results of the LLE indicated that both the swarm exists in the 'Edge of chaos,' displaying complex dynamical behavior. The final system state studied for all the cases indicated

binary final states for some cases in Table.3.1. Multiscale Entropy(MSE)[73, 74] was also evaluated from the center of mass time-series to study the complexity of the interacting swarms. The MSE results (Fig. 3.2) revealed a greater degree of randomness for the Defenders compared to Attackers due to the nature of the former's role in the simulation. Overall, the maximum value of MSE obtained was ~1.40 for both the Attackers and Defenders combined. The average MSE for the Attackers and Defenders is 1.04 and 1.25, respectively when calculated over time scales from 1 to 20. As a result of the medium degree of randomness of the non-linear center of mass time-series for both classes of agents, the prediction of the time-series data was plausible.

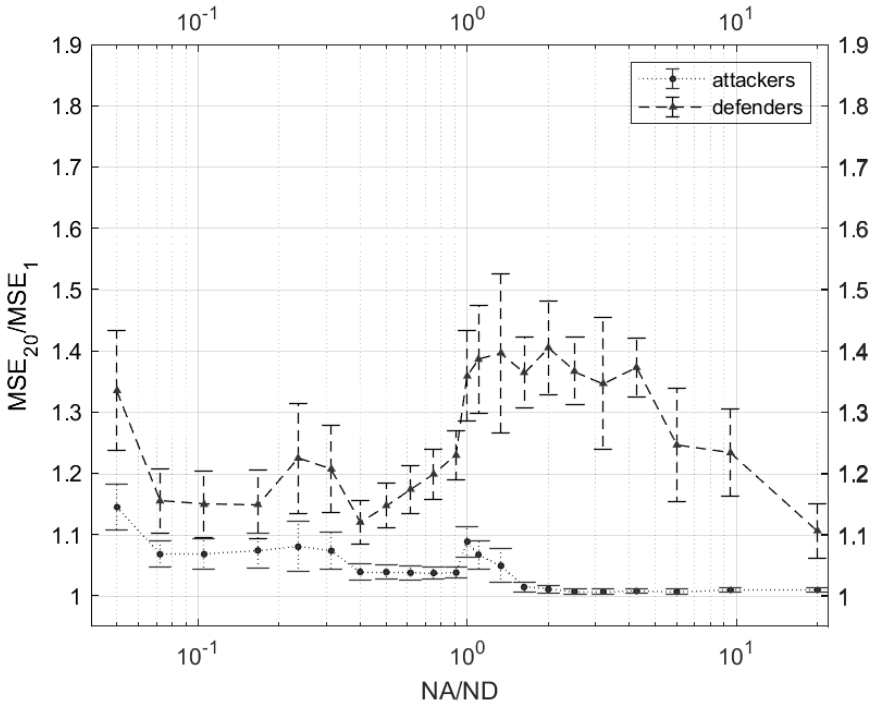

Figure 3.2: The ratio of Multiscale Entropy calculated at scale 20 and scale for the center of mass time series for Attacker and Defender swarm, respectively

3.2.2 Echo State Networks

In a general Echo State Network (ESN) shown in Fig.3.2, an input vector (training data) u(t) with K units (or nodes) is fed into a dynamic ‘Reservoir’ with N units (or nodes). The

Reservoir is then coupled to an output layer y(t) with L units(or nodes)[53]. It is assumed that the Reservoir receives an input at discrete time t and is combined with the reservoir state to produce its output $t + \partial t$.

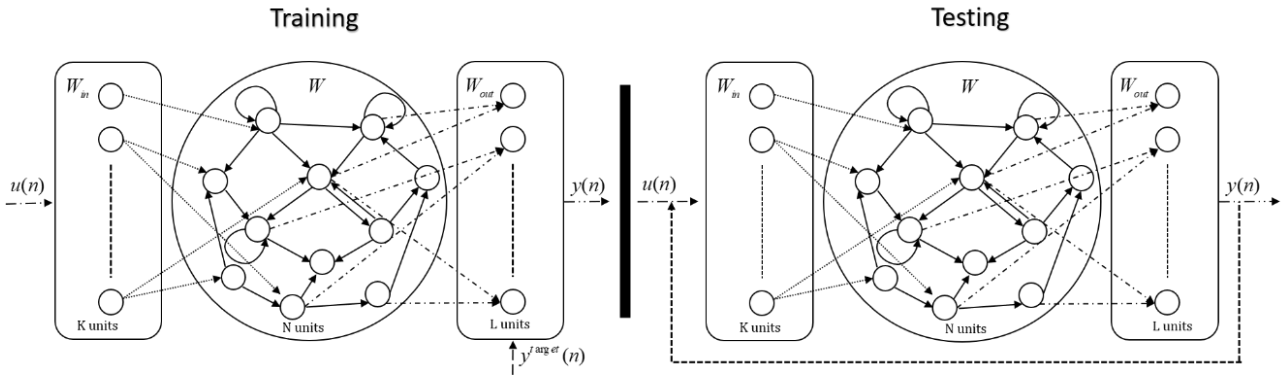

Figure 3.3: Training and Testing of an Echo State Network.

Echo State Networks typically use an RNN with leaky-integrated discrete-time continuous value units[75]. The equations of update used in this study are obtained from[59, 76] and are given by the following equations:

$$\tilde{x}(n) = \tanh(W^{in}[1; u(n)] + Wx(n-1)) \tag{3.5}$$

$$x(n) = (1-\alpha)x(n-1) + \alpha\tilde{x}(n) \tag{3.6}$$

Where $x(n) \in \mathbb{R}^{N_X}$ represents neuron activations of the Reservoir and $\tilde{x}(n) \in \mathbb{R}^{N_X}$ is its update at every time step n. The tanh() function is used as the sigmoid wrapper for the neurons. $\alpha \in (0,1]$ is the leaking rate, [;] represents a vertical vector (or matrix) concatenation $W^{in} \in \mathbb{R}^{N_X \times (1+N_U)}$ and $W^{N_X \times N_X}$is the input weight matrix and recurrent weight matrix, respectively[76]. Once the Reservoir is trained, the output weight matrix is generated using linear regression. The linear readout layer is defined as,

$$y(n) = W^{out}[1; u(n); x(n)] \tag{3.7}$$

Where $y(n) \in \mathbb{R}^{N_y}$ represents the network output $W^{out} \in \mathbb{R}^{N_y \times (1+N_u+N_X)}$ is the output weight matrix, and [;] represents a vertical vector concatenation. The network output weight

matrix is learned by comparing the network output y(n) with the target output $y^{target}(n)$. This task is achieved by minimizing the Root Mean Square Error (RMSE) between the network output and the target output[76, 77].

$$E(y, y^{target}) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{T}\sum_{n=1}^{T}(y_i(n) - y_i^{target}(n))^2} \tag{3.8}$$

In Eqn. (3.10), N represents the total dimension of the output matrix. The operation of the ESN, as defined by[36], can be briefly summarized by the following steps:

I. Generation of a random reservoir with a given input weight matrix W^{in}, recurrent weight matrix W, and a given leaking rate α.

II. The training input u(n) drives the network, and reservoir activation units x(n) are collected.

III. A linear readout layer obtains the network output. The output weights W_{out} are computed by minimizing the error between the network output $y(n)$ and the target output $y_I^{target}(n)$.

IV. Finally, compute output in the prediction phase using the trained network on the recycled data (i, e, the data obtained in the previous time-step) in case of generative mode or a limited component time-series in predictive mode.

To produce a highly accurate network capable of predicting the output data, the hyperparameters of the RC-ESN viz, spectral radius scaling factor, leaking rate, probability of non-zero connections, and regularization coefficients must be tuned. This task was achieved by performing a comprehensive uniform grid search essential to producing a well-trained network capable of predicting the highly transient dynamical behavior of the Adversarial Swarm system.

The readouts from the ESN in Eqn. (3.7) can be rewritten using matrix notation as:

$$Y = W^{out}X \tag{3.9}$$

Where $Y \in \mathbb{R}^{N_Y \times T}$are is$y(n)$ and $X \in \mathbb{R}^{(1+N_U+N_X)\times T}$ are all $[1; u(n); x(n)]$ produced by presenting the Reservoir with $u(n)$, both collected into respective matrices by concatenating the column-vectors horizontally over the training period n=1,……, T. For notational simplicity, X is used instead of $[1;\ U;\ X]$

The optimized weights W^{out} minimize the RMSE between $y(n)$ and $y^{target}(n)$ are obtained by solving the following set of linear equations, which are overdetermined in nature.

$$Y^{target} = W^{out}X \tag{3.10}$$

Where $Y^{target} \in \mathbb{R}^{N_y \times T}$a stable solution to Eqn. (10) is obtained by Ridge Regression, also known as Linear Regression[78] with Tikhonov Regularization.

$$W^{out} = Y^{target}X^T(XX^T + \beta I)^{-1} \tag{3.11}$$

The output weights W^{out} in equation (11) are found out by minimizing the RMSE in Eqn. (8) by the following equation:

$$W^{out} = \text{argmin}\frac{1}{N_y}\sum_{i=1}^{N_y}(\sum_{i=1}^{T}(y_i(n) - y_i^{target}(n))^2 + \beta||W_i^{out}||^2) \tag{3.12}$$

Where w_i^{out} is the ith row of W^{out} and ||.|| stands for the Euclidian norm. The above equation's objective function includes a regularization term $\beta||w_i^{out}||^2$ penalizing large sizes W^{out} to the square error between the predicted data $y(n)$ and the target data$y^{target}(n)$. The regularization term aims to bring about a compromise between training errors and output weights.

3.2.3 Echo State Networks (ESN): Configuration 1

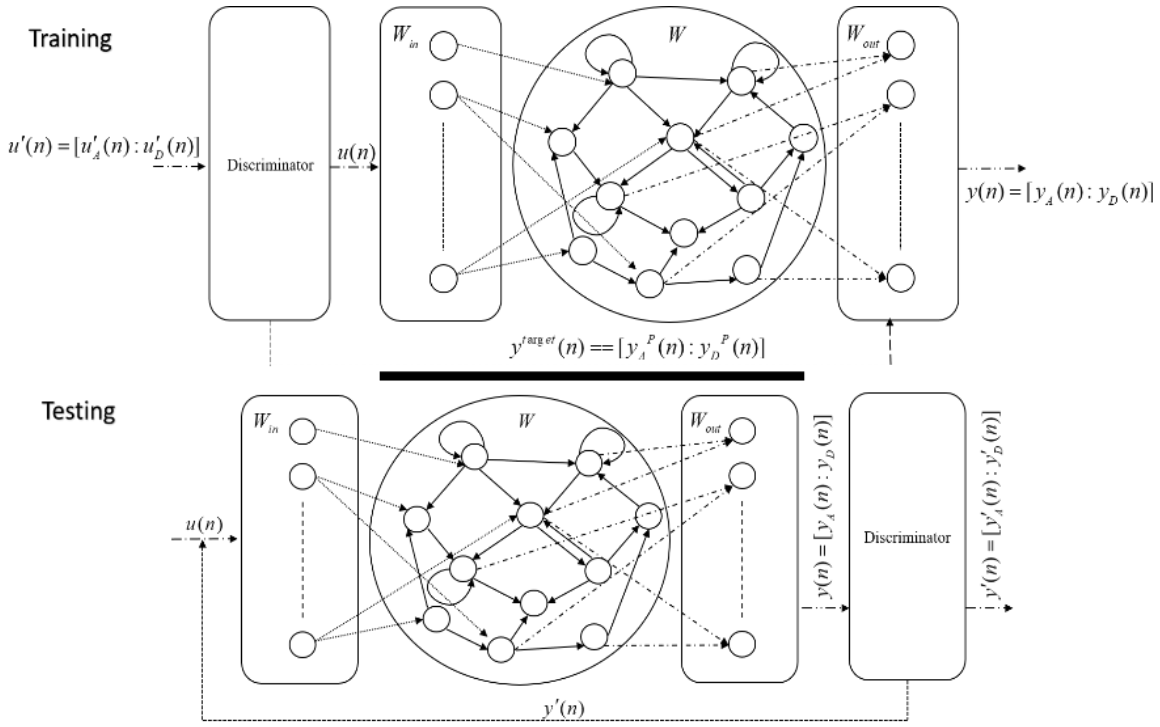

Figure 3.4: Training and Testing of ESN Config#1.

In this configuration, a single high-dimensional reservoir will be used to predict the dynamics of the individual agent (Fig. 3.3). The spatial time-series data of the position vectors obtained from the Attackers and the Defenders from [79] will form the basis of the training and the testing set. The position vectors for all the agents of the Attacker and the Defender swarm, respectively, are denoted by $u'_A(n)$and $u'_D(n)$. The position vectors of the agents of both the swarm are further concatenated and presented as a single input to the Reservoir by means of the vector $u'(n)$. This vector is essentially the position vector for all the agents taken together as one high-dimensional concatenated vector. During the training phase, the data is first passed through with an external module called the 'Discriminator,' which preprocesses the training data for input into the ESN. The preprocessor's main aim is to modify the position vector data for compromised agents to maintain uniformity in the dimension of the data input to the ESN. In the prediction

phase, the preceding time step is used to predict every successive time step. In the prediction phase, the 'Discriminator' is used to enforce the simulation end criteria as presented in [79] to serve three principal purposes,-first to stop the prediction in case the outcome of the simulation is determined (i,e to declare whether the Attackers or the Defenders emerge as the dominant party); second to post-process the predicted output by implementing the agent collision criteria in comparing the distances between all Attacker and Defender agents respectively to the agent compromise criteria and third, if any agent collides, post-process the predicted data further for input into the RC-ESN for the prediction of the subsequent time-step as it is operating in the generative mode.

3.2.4 Echo State Networks (ESN): Configuration 2

In the second configuration, two RC-ESNs work in parallel, wherein each is responsible for handling the Attackers and Defenders swarm data, respectively (Fig. 3.4). In both these cases, the ESN will be coupled with an external 'Discriminator' that will play the same role as that first configuration, as discussed in the preceding paragraph. The input to each parallel Reservoir is the same $u'(n) = [u'_A(n): u'_D(n)]$. In contrast, the output is only trained against ${y_A}^{target}(n)$ and ${y_D}^{target}(n)$ which is the concatenation of position vector of all the Attacker and the Defender agents, respectively. The target data is obtained from the Discriminator (See Fig. 3.4,${y_A}^{P}(n)$ & ${y_D}^{P}(n)$) as it is a part of the training data; the Discriminator would also ensure that the target data is also preprocessed. The common input ensures that the respective ESNs can learn the interaction of the Adversarial Swarms while predicting the dynamics of each swarm during the prediction phase. The ESN in this configuration will be operated in generative mode (Fig.4 with feedback from the previous time-step). The LLE of the respective Attacker and Defender swarm were calculated from the reservoir of the ESN, which was essentially an N^{th} dimensional discrete-

time dynamical system. The equations for the evolution of the tangent map of Eqn.5 and a set of n mutually orthogonal tangent vectors were evaluated. The LLE was computed from the among n Lyapunov exponents of the dynamical reservoir of ESN in the testing phase, as shown in Fig.3 based on QR decomposition. The details of this method can be found in Verstraeten[40], and Pathak[53]; The LLE of the predicted data in the ESN will be compared to that obtained from the simulation in[79] employing attractor phase space reconstruction[70] and Wolf's algorithm[71].

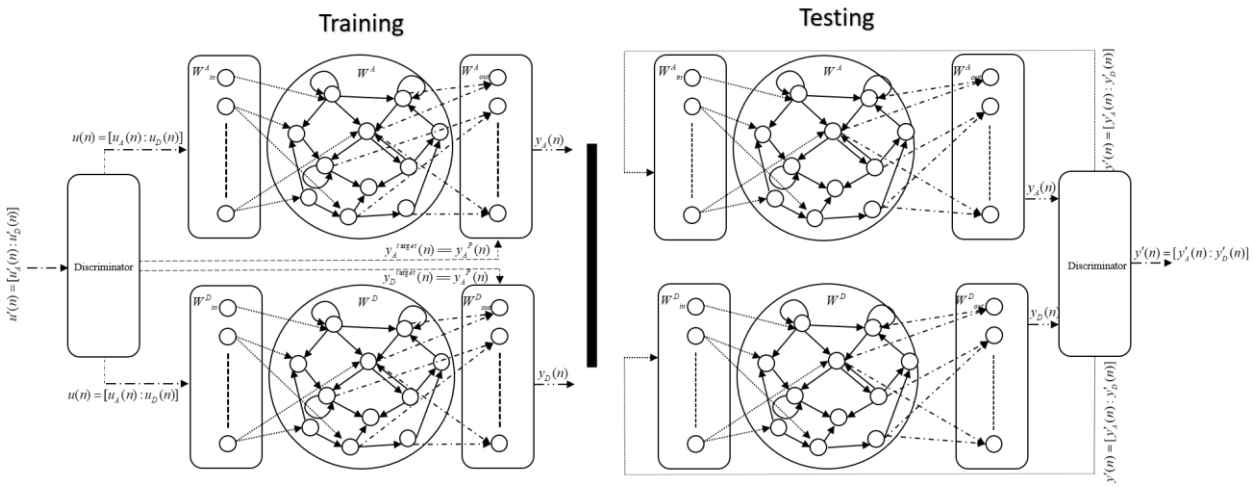

Figure 3.5. Parallel ESN architecture (configuration #2).

3.4 Verification and validation of ESN

The Echo State Network was verified and validated by training the classical Lorentz time series[80], a benchmark example of a chaotic time-series. The ESN is trained on the data obtained by solving the Lorentz system$\sigma = 10, \rho = 100, \beta = 8/3$ for initial conditions $(x = 2.65, y = 2.0, z = 20)$ from 0 to 125 using a fixed time-step ode45solver. A time-step of 0.0001 was considered. The Lorentz time-series was used to train an ESN from time 0 to 100, and the system was predicted from 100 to 125. The prediction is made in the so-called generative mode of operation, where the predicted data from the previous time step was used to drive the ESN to get

the prediction in the subsequent time-step, as shown in Figure 4. A comprehensive hyperparameter search of the size of the Reservoir, leak rate and probability of non-zero connections, and the regularization coefficient was conducted to optimize the ESN on the training data only.

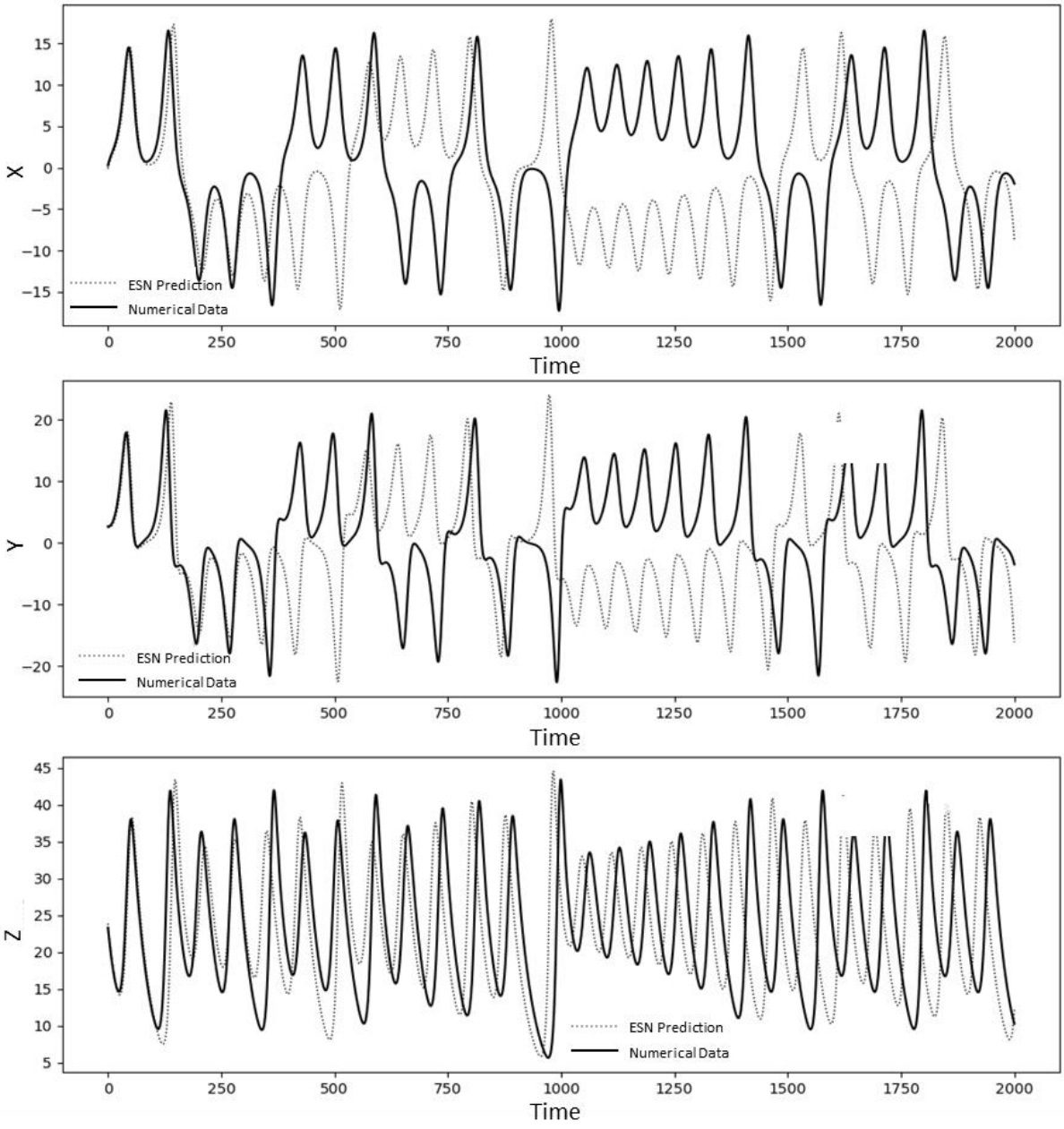

Figure 3.6. Verification and Validation of ESN.

The best Normalized Mean Square Error (MSE) of the prediction is 0.1187. The hyperparameters of the ESN used for predicting the Lorentz time series are reservoir nodes 750, leak rate 0.1, probability of non-zero connections 0.1, and regularization coefficient 1e-05. It is observed that the ESN was able to generate correct predictions initially in the short term. After the initial portion of the graph, as indicated in Fig. 6, the prediction deviates as the error grows exponentially as the output at every time-step is obtained by feeding in the output at the preceding time-stepwise in the generative mode of operation. Nonetheless, the long-term dynamics of the system resemble that of the original system. These results are in affirmation of the results found in the literature [53]. The ESN code developed for the current work can be thus considered to be verified and validated. The ESN's ability to qualitatively predict a dynamical system will be exploited in the later sections of this paper.

3.5 Results and Discussion

A total of 22 cases of the Adversarial Swarm was studied in [23] as presented in Section 3.3 of this paper. Among these 22 cases, 3 primary cases were chosen to predict using the Echo State Network. These 5 cases were deemed enough to prove the robustness and the reliability of the ESN to achieve a hybrid prediction for the Adversarial Swarm System. A total of 2 ESN configurations were considered, as discussed before. The predictions from each configuration are discussed comparatively for each case respectively in the preceding subsections.

Three primary cases were chosen among the 22 cases of the numerical experiment are Case 1 with 5 Attacker and 100 Defenders, Case 12 with 50 Attackers and 50 Defenders, and Case 22 with 5 Defenders and 100 Attackers. The cases simulate three critical situations that may arise in the interacting Adversarial Swarms based on their population: a vast number of Defenders and relatively few Attackers, Attacker and Defender numbers comprising respective swarms are

equal Attacker swarm vastly outnumber the Defender swarm. These cases will be used to establish the robustness and the reliability of the hybrid ESN model discussed before.

A comprehensive hyperparameter search was conducted for each of the cases in both configuration 1 and 2, respectively, and the best combination of hyperparameters was found out by minimizing the NRMSE in each case, respectively. 44% of the available data was used to train the ESN, and the rest, 56%, was used as the testing set by the ESN in the generative or autonomous mode of operation; this ratio was empirically determined as the best ratio over repeated numerical experiments. The results for every case in both the configurations are depicted in Tables 3.1 and 3.2, which are explored on a case-by-case basis in the subsequent subsections.

In Config 1, a single high dimensional hybrid ESN was used to achieve the prediction for the Adversarial Swarm System as a whole. The spatial time-series data for individual Attacker and Defender agents of the Attacker and Defender swarm was used to train the network. During the testing phase, the outputs of all the agents were also considered individually. The center of mass time-series was generated by taking the mean of the 2D spatial coordinates of active agents at every time-step for the Attackers and the Defenders, respectively. The ESN was trained on the first

44% of the available time-series data of every individual agent in both swarms respectively; the trained ESN is subsequently used for prediction by using the prediction at every time-step as the driving signal to predict the subsequent time-step, the Discriminator is used to post-process the output prediction at every time-step before it is then fed back into ESN for the prediction at the next time-step. The purpose of the discriminator is to enforce the agent

collision and simulation end criteria on the prediction at every time-step, respectively, as outlined in Gupta et al[23]. The center of mass time-series of the prediction is calculated by averaging the 2D spatial coordinates of active agents at every time-step. The center of mass time-series prediction results can be found in Fig.3.7, the network is trained until the left of the vertical line, and the prediction starts from the right of the vertical line; this convention is followed for other time-series prediction plots in this paper.

Table 3.2: Configuration 1 ESN hyperparameters and NRMSE

	Case#			
	1	**12a**	**12b**	**22**
Outcome	Defenders Win	Defenders Win	Attackers Win	Attackers Win
Simulation End Time	100	11.48	11.27	0.604
Prediction End Time	100	11.48	5.9	0.604
Number of Reservoir Nodes	310	100	310	100
Probability of non-zero connections	0.05	0.1	0.1	0.05
Leak Rate	0.05	0.1	0.1	0.1
Regularization Coefficient	1.00E-05	1.00E-05	1.00E-05	1.00E-04
NRMSE	0.11	0.18	0.08	0.79

Table 3.3: Configuration 2 ESN hyperparameters and NRMSE

	Case#			
	1	**12a**	**12b**	**22**
Outcome	Defenders Win	Defenders Win	Attackers Win	Attackers Win
Simulation End Time	100	11.48	11.27	0.604
Prediction End Time	73	11.48	8.1	0.4905
Number of Reservoir Nodes	310	100	310	100
Probability of non-zero connections	0.05	0.1	0.1	0.05
Leak Rate	0.05	0.1	0.1	0.1
Regularization Coefficient	1.00E-05	1.00E-05	1.00E-05	1.00E-04
NRMSE	0.05	0.23	1.7	0.11

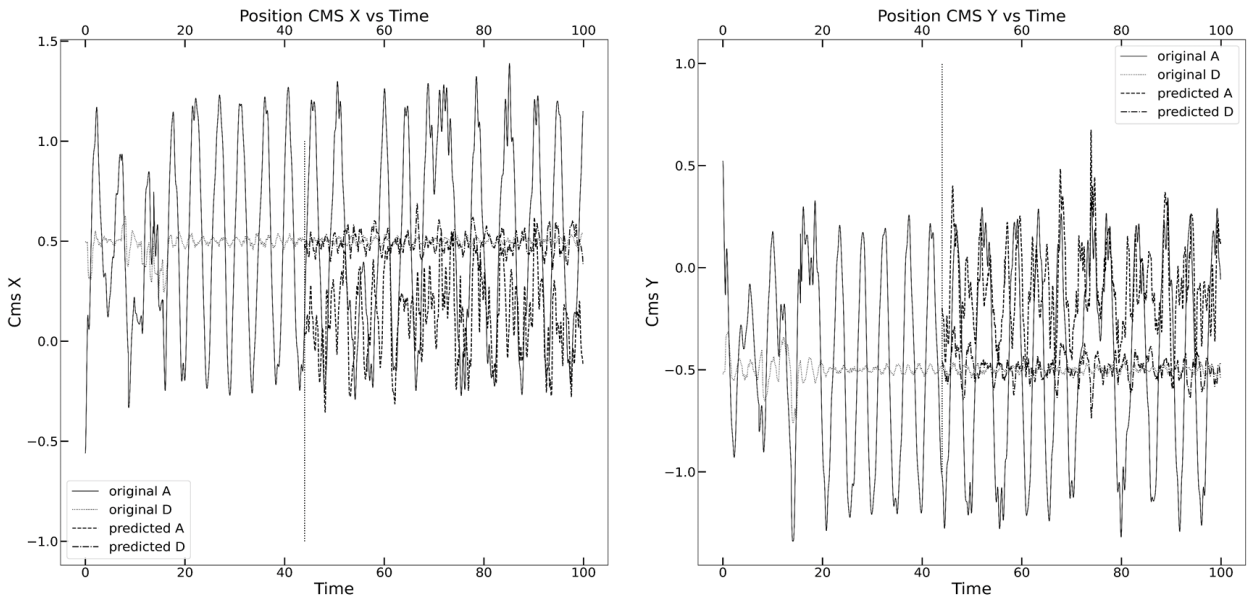

Figure 3.7. Hybrid ESN Config 1 prediction for case1

Initial very short-term prediction indicates a good match, thereafter the error grew exponentially, as the prediction deviates considerably from the original data. However, it was observed that the prediction is qualitatively similar to the original data. In case the number of agents of the Defender swarm is 100 whereas the number of agents in the Attacker swarm is 5, thus case 1 is a trivial case where the outcome of the interaction can easily be guessed; as indicated in Table 3.3, the simulation ending time for case 1 is 100, the same end condition was met exactly by the ESN in Config 1 in Fig. 3.7 (right). The prediction indicated that the ESN was able to capture the dynamics of the system to a large extent. The NRMSE obtained was 0.11 (Table 3.1). The same prediction task was also conducted by Config 2.

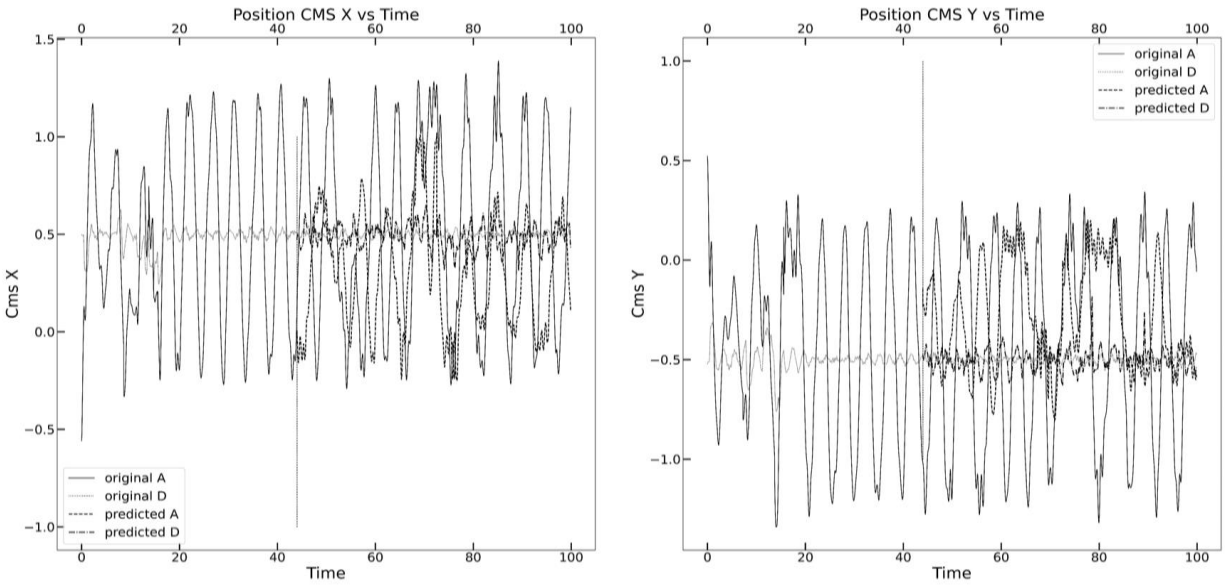

Figure 3.8. Hybrid ESN Config 2 prediction for case1

The case 1 data was also used to train the hybrid ESN in Config 2, the training and the prediction was obtained in the same way as that of Config 1, the results of the predicted center of mass time-series were depicted in Fig. 3.8, it was observed that the prediction is better compared to Config 1. In Config 1, a single high dimensional ESN was used to achieve the prediction task; the hybrid ESN in Config 1 could not effectively capture the dynamics of the interacting adversarial system as these two systems are essentially independent to each other, and their interaction is external in nature. In Config 2, however, two ESNs assigned to each agent are trained with a common input that is the Attacker and Defender data taken together and are trained against the Attackers and Defenders separately. This process is conducted in parallel as depicted in Fig. 5 using mpi4py across two parallel processes, and a third process is assigned to 'Discriminator', which preprocessed the input during training and post-processes(enforcing simulation end criteria) the output at every time step to create the driving signal for predicting the subsequent time-step.

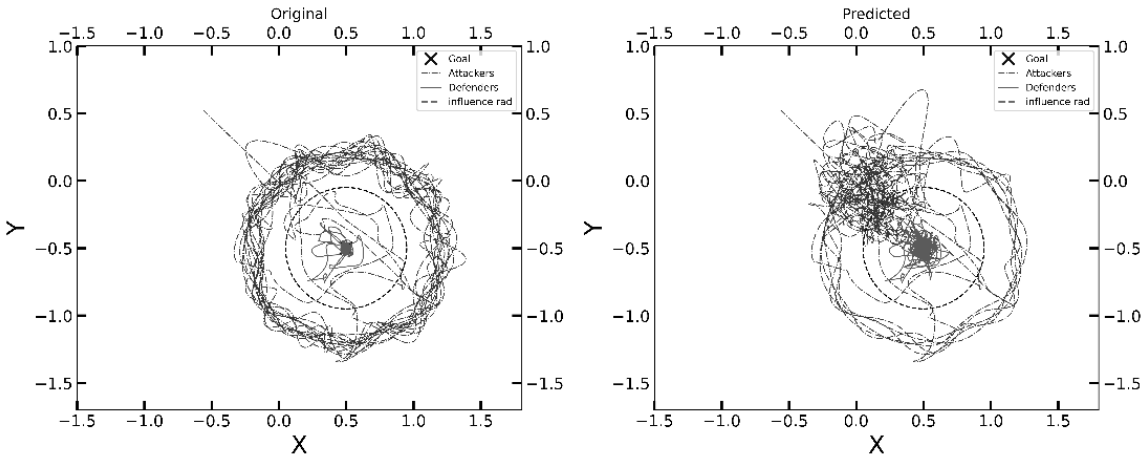

Figure 3.9. Visualization in 2D space: original and predicted center of mass time-series in Config 1.

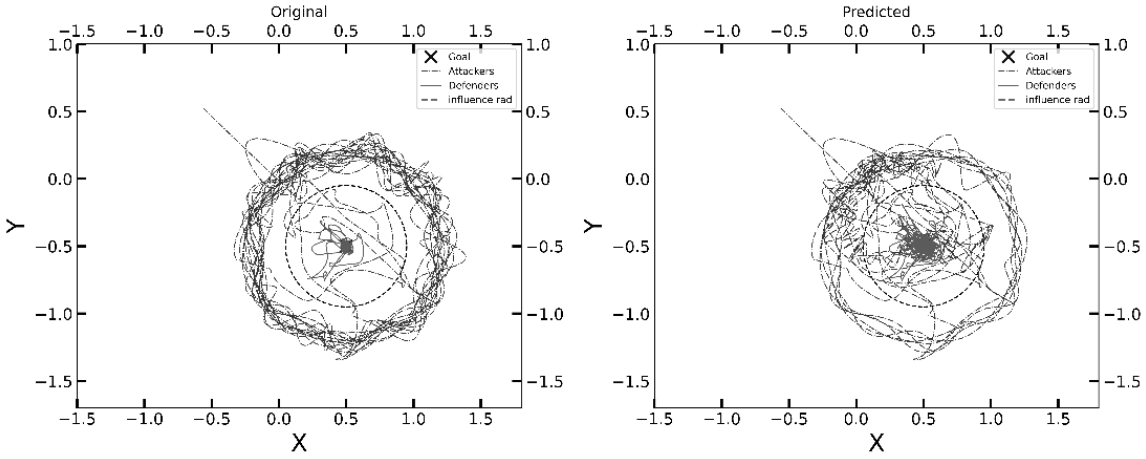

Figure 3.10. Visualization in 2D space: original and predicted center of mass time-series in Config 2

The interaction between the two swarms can be visualized in 2D space by plotting the 2D spatial coordinates of the center of mass time-series and evolving them over time; Fig. 9 and 10 represent the 2D phase space visualization of the Attackers and the Defenders in Config 1 and Config 2 respectively. It can be observed that the attractors in the original and the predicted plots are similar to each other for both Config 1 and Config 2. However, for the reasons mentioned in the preceding paragraph, it can be concluded from Fig 3.9 and 3.10 that the hybrid ESN in Config 2 was effectively able to capture the dynamics of the interacting swarm system when compared

to the ESN in Config 1, especially this is true for the Attackers, and it can be easily observed that for case 1 the Attackers seemed to clutter near the Goal for Config 1 when compared to the semi-periodic motion of the Attackers observed for Config 2.

The hybrid ESN in Config 1 and Config 2 was similarly used to predict the remaining 12 and 22. In case 12, the number of Attacker and Defender agents are equal to 50 each; in this case, however, there can be two outcomes, either the Defenders or the Attackers emerge as the dominant swarm. The binary outcomes in case 12 was found out by conducting a Monte Carlo Simulation of case 12 in [23] Gupta et al. For the current study, these two distinct outcomes were considered distinctly and were labeled as 12a (where Defenders emerge dominant) and 12b (where Attackers arise dominant), these two cases are investigated by the hybrid ESN setup in config1 and config2 respectively.

The hybrid ESN setup was trained in Config 1 mode using the same procedure as described in the preceding subsections for case 12; the prediction of the center of mass time series was depicted below. In case 12, however, binary final states existed, which means either the Attackers or the Defenders emerged as the dominant swarm in the 1000 trial Monte Carlo Simulation conducted in Gupta et al. [23]. Two distinct trials where the Defenders and Attackers occurred separately as the dominant swarm was considered two test cases (12a and 12b), respectively. This approach was considered as it is imperative that the ESN is able to predict the final state of the system correctly. The hybrid ESN in Config 1 and 2 was used to train the data for case 12a (Defenders won) in Table 1. The ESN in both the configurations was correctly able to predict the final outcome of the simulation; that is, the Defenders emerged as the dominant swarm both in the final simulation and the prediction as all the Defender swarm was able to compromise the agents of the Attacker swarm ultimately. The NRMSE of this prediction in Fig 11 was 0.18.

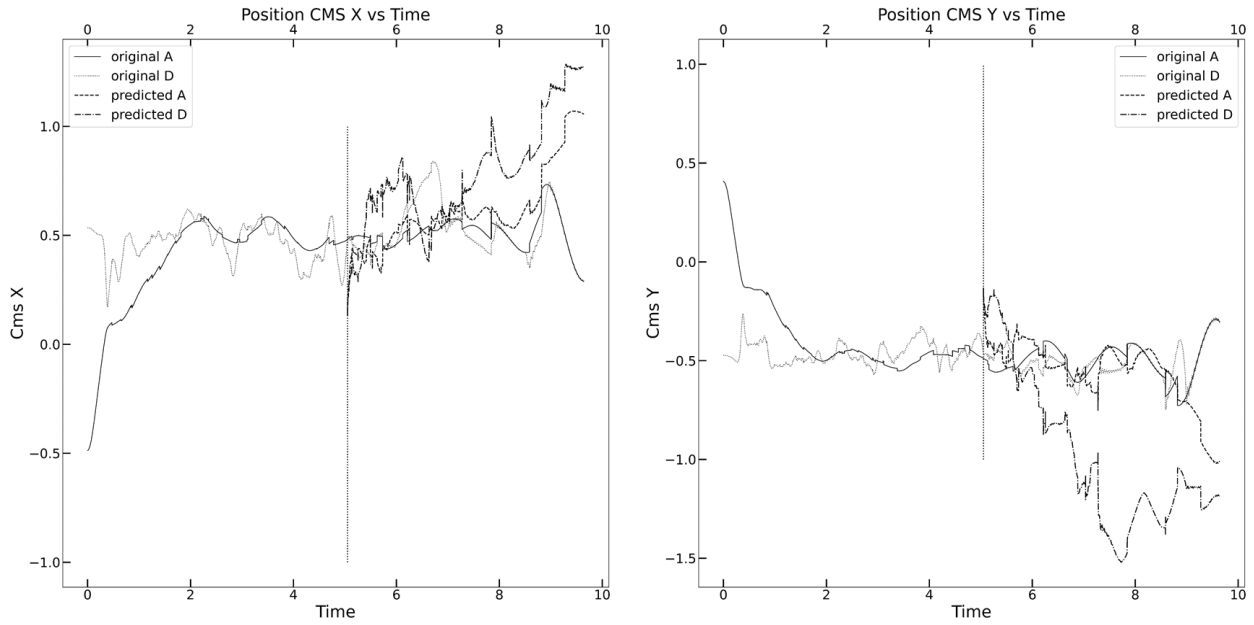

Figure 3.11. Hybrid ESN Config 1 prediction for case12a (Defenders emerged dominant)

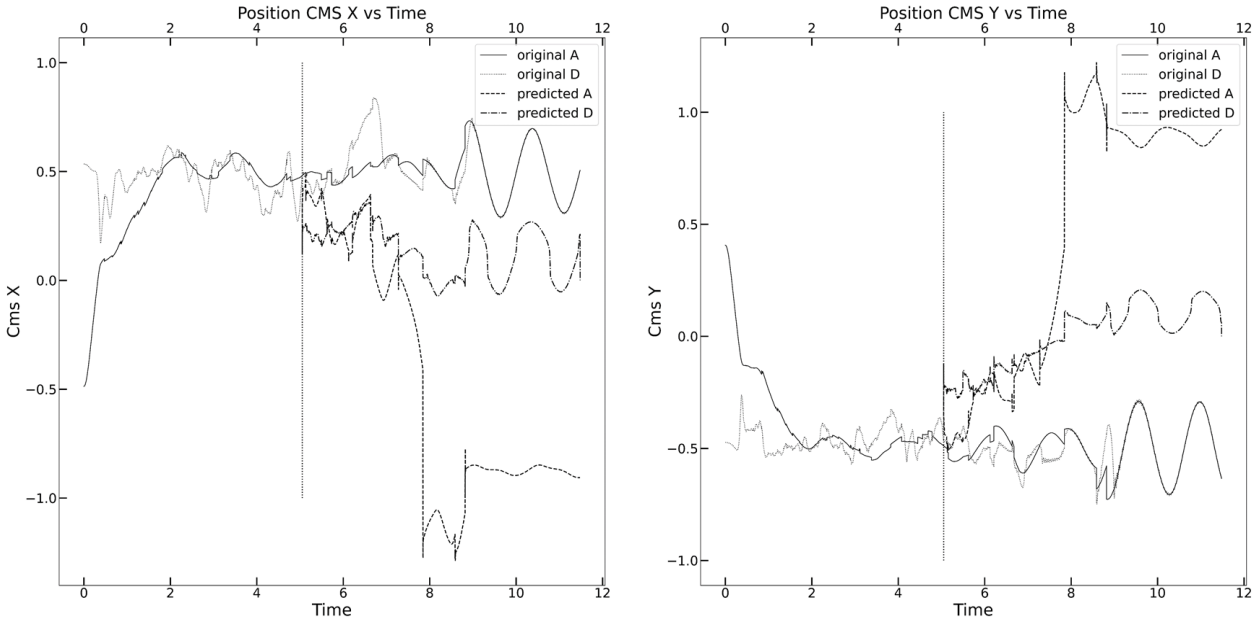

Figure 3.12. Hybrid ESN Config 2 prediction for case12a (Defenders emerged dominant)

The data in case 12a was also used to train the hybrid ESN setup in Config 2 mode. The results of the center of mass time-series predictions are depicted in Fig.3 .12; the ESN was correctly able to predict the final outcome of the simulation; that is, the Defenders emerged as the dominant swarm. The NRMSE for Config 2 was 0.23 when compared to 0.18 of Config 1. The prediction

plots for Config 2 reveals that the ESN in Config 2 performed moderately when capturing the dynamics of the Attackers when compared to the Defenders. Overall, this behavior may be attributed to the high degree of randomness in the data for case 12, as the average MSE for the Adversarial Swarm system in case 12 (see Fig 3.2) was much higher when compared to the other cases.

In case 12b, both Config 1 and Config 2 were used for the prediction task using the same training and testing prediction as indicated before. The hybrid ESN was correctly able to predict the final outcome of the simulation in both configurations. The prediction of the center of mass times series for case 12b using Config 1 is depicted in Figure 13; it was also observed that the Attacker swarm is able to engage with the Defender swarm and intercept the Goal quite early on the prediction. The prediction was discontinued after the Goal was unreasonable to continue the prediction thereafter. The original simulation ended at 11.27, where the simulation is called off at 5.9; there is a time difference of around 49%. It is clear from Fig. 3.13 that the hybrid ESN in Config 1 cannot capture the dynamics of the swarm effectively and s able to predict the final state of the system partially.

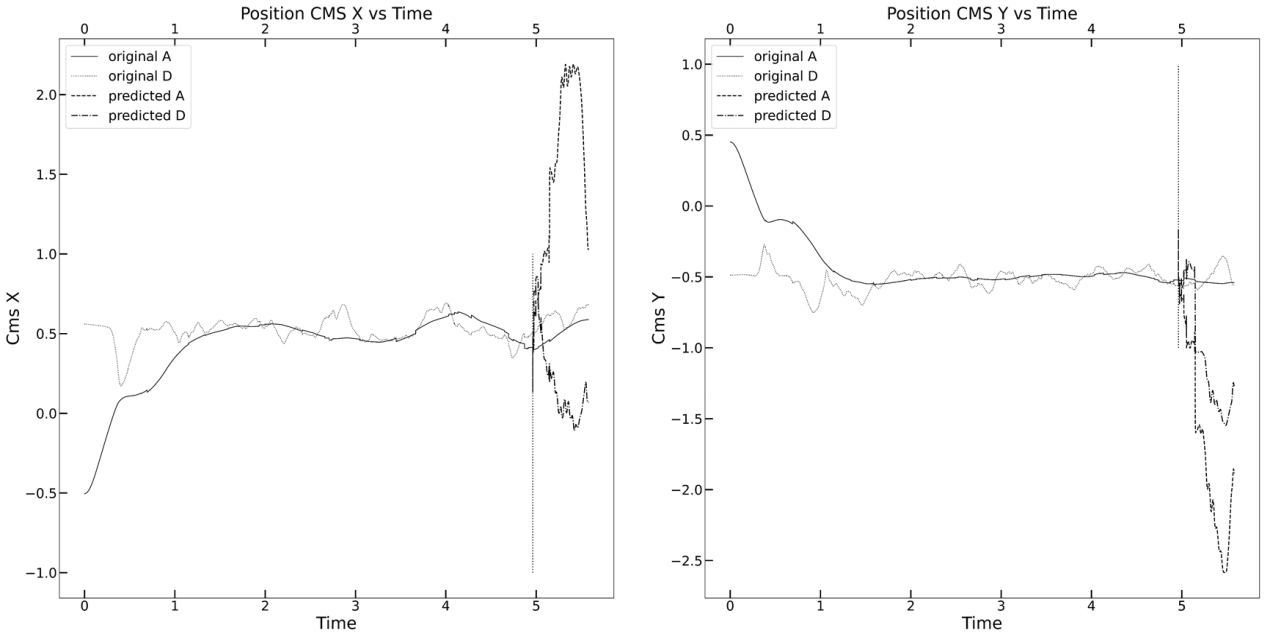

Figure 3.13. Hybrid ESN Config 1 prediction for case12b (Attackers won)

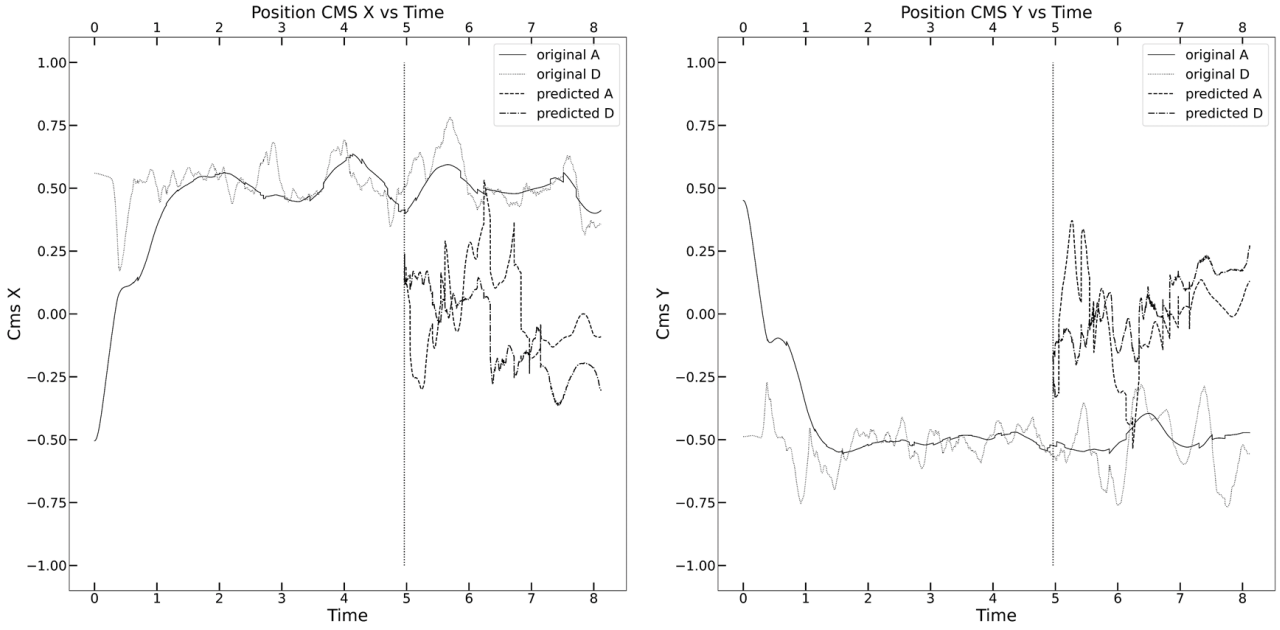

Figure 3.14. Hybrid ESN Config 2 prediction for case12b (Attackers won)

However, in the case of Config 2, the hybrid ESN performed much better in capturing the dynamics of both the Defenders and the Attackers, respectively (see, Fig.3.14). The simulation end time is 8.1 compared to 11.27 for the original simulation, which was much better when compared to Config 1.

In case 22, the number of agents making up the Attacker swarm was 100, whereas the number of agents making up the Defender swarm was just 5; the outcome of such a simulation is trivial as the Defenders are heavily outnumbered by the Attackers. In both the Config(s) of the hybrid ESN, it was observed that the final outcome is predicted correctly. However, Config 2 indicated better performance over Config 1 the NRMSE(s) are 0.11 and 0.79, respectively. The MSE of case 22 is lower when compared to the other cases (see, Fig 3.2). Thus ESN in Config 2 was observed to be better than Config 1, although there was minimal difference between the total predicted time of Config 2 when compared to Config 1.

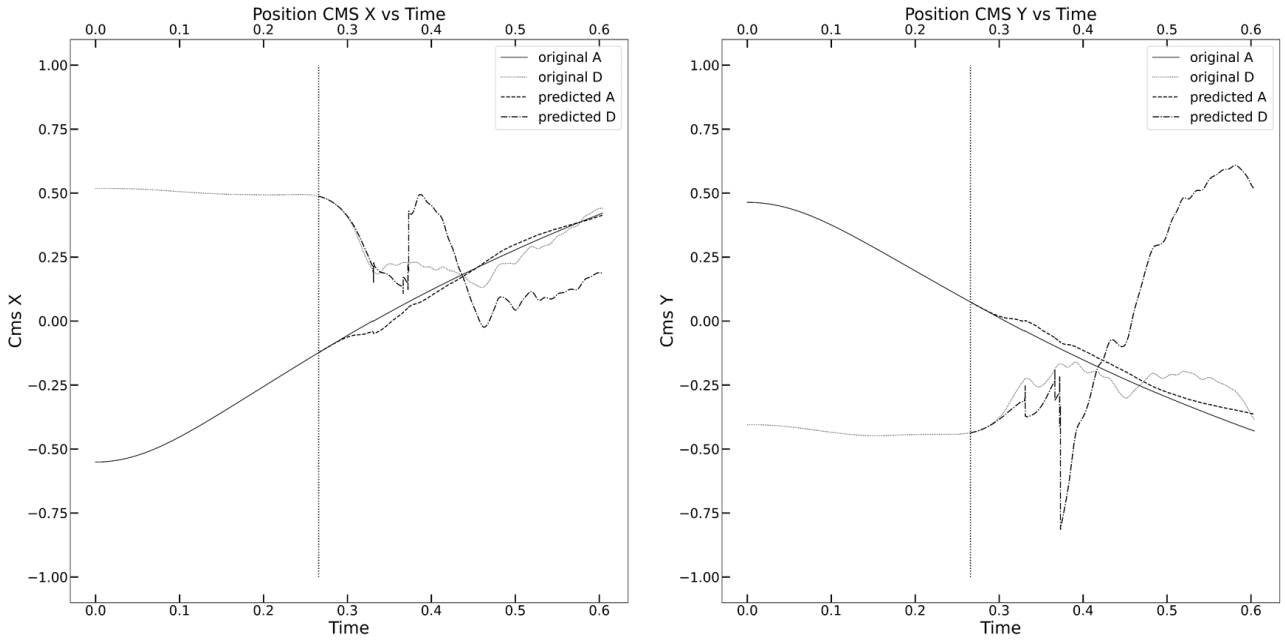

Figure 3.15. Hybrid ESN Config 1 prediction for case 22

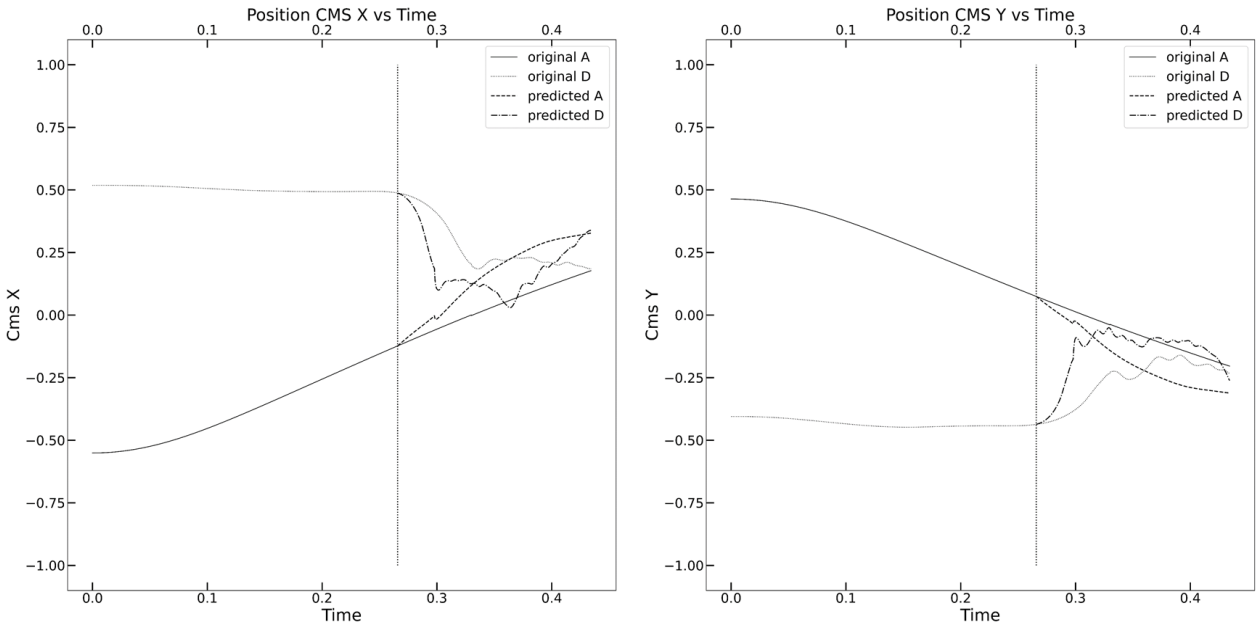

Figure 3.16. Hybrid ESN Config 2 prediction for case 22

Table 3.4: Lyapunov Exponent comparison

	Wolf's method		**ESN**	
	A	D	A	D
Case1	2.00E-04	5.00E-03	-0.013	-0.0015
Case12	-2.00E-03	1E-02	-0.01	-0.01
Case22	3.00E-04	4.00E-02	5.00E-02	2.00E-02

Lastly, the Largest Lyapunov Exponent (LLE) to the center of mass time-series was calculated for each case respectively using Wolf's Algorithm[71] and is compared to the LLEs obtained from for each of the parallel ESNs used in Config 2 that were assigned to the Attackers and the Defenders respectively. The LLE from the ESN is obtained by considering the Reservoir in the ESN as a discrete-time D_r dimensional system; the detailed method of LLE evaluation can be found in [53] . The LLE obtained from the hybrid ESN in Config 2 only is compared to that of the center of mass times series and can be found in Table 3.3; overall, the results seem to be in good consideration. The ESN appears to be near the 'Edge of Chaos', which is the same case for the majority of the data obtained from the Adversarial Swarm system.

3.6 Conclusion

In the current work, a hybrid ESN was developed in two configurations, namely Config 1 and Config 2 to predict the dynamics of an interacting Adversarial Swarm system. The swarm consists of two distinct agents, namely the Attackers and the Defenders, who have conflicting objectives in 2D space. The Defender's main task was to protect the Goal by swarming around it around a perimeter termed as the sphere of influence, and the Attackers are constantly trying to intercept the Goal. An Echo state network was first verified and validated against the classical Lorentz system. The results indicated that the ESN was successful in predicting a qualitatively similar dynamical behavior.

Two hybrid ESN prediction methods were developed-Config 1 and Config 2, respectively, which were used for predicting Adversarial Swarm dynamics for three distinct cases. These cases were made up of a varied number of Attackers and Defenders, respectively. The three distinct cases were chosen keeping in mind scenarios of interest that may arise in various engineering applications.

The hybrid ESN in Config 1 consisted of a very high dimensional ESN where both the Attacker and Defender agent are taken as common input via the Discriminator during training. During testing, the ESN predicted the Spatio-temporal data for every individual agent making up the Attacker and the Defender swarm. This output was then fed into the Discriminator, wherein agent compromise criteria and goal breach criteria are enforced during the prediction phase.

In Config 2, two parallel hybrid ESNs worked in conjunction with a common Discriminator. Each ESN was assigned to the Attackers and Defenders, respectively. This configuration was developed as the Attacker, and the Defender swarm interaction was extrinsic in nature and was modeled by the inter swarm forces as described in the introductory sections. The parallel hybrid

ESN had a common input, the Spatio-temporal data for both swarms, to map the interaction of the swarms, whereas the output for each swarm was trained against the individual spatio-temporal data of individual swarms. Thus, the idea was to force the ESN to learn the dynamics of individual swarms, including the goal breach criteria, agent compromise criteria, and simulation end criteria.

The following main conclusions can be drawn from the current work:

- ESNs are overall capable of learning the dynamics of highly nonlinear agent-based Adversarial Swarm System
- Among the two configurations developed in this work, Config 2 showed better overall performance when compared to Config 1 to achieve the prediction task.
- Config 2 was able to achieve a better prediction as it consisted of two parallel independent ESNs, which were successfully able to learn the dynamics of individual swarms, which are essentially independent of each other and are only coupled by external rules of engagement.
- The LLEs obtained from the ESNs and phase space reconstruction were in good agreement with each other, further implying the success of the ESN to map the interacting swarm dynamics successfully.

REFERENCES

1. Selvi, V., and Umarani, R. "Comparative analysis of ant colony and particle swarm optimization techniques," *International Journal of Computer Applications* Vol. 5, No. 4, 2010, pp. 1-6.

2. Carere, C., Montanino, S., Moreschini, F., Zoratto, F., Chiarotti, F., Santucci, D., and Alleva, E. "Aerial flocking patterns of wintering starlings, Sturnus vulgaris, under different predation risk," *Animal Behaviour* Vol. 77, No. 1, 2009, pp. 101-107.

3. Calovi, D. S., Lopez, U., Ngo, S., Sire, C., Chaté, H., and Theraulaz, G. "Swarming, schooling, milling: phase diagram of a data-driven fish school model," *New journal of Physics* Vol. 16, No. 1, 2014, p. 015026.

4. Perea, L., Gómez, G., and Elosegui, P. "Extension of the Cucker-Smale control law to space flight formations," *Journal of guidance, control, and dynamics* Vol. 32, No. 2, 2009, pp. 527-537.

5. Edwards, S. J. "Swarming on the Battlefield: past, Present, and Future." RAND NATIONAL DEFENSE RESEARCH INST SANTA MONICA CA, 2000.

6. Al-Rifaie, M. M., Aber, A., and Raisys, R. "Swarming robots and possible medical applications," 2011.

7. Bailey, I., Myatt, J. P., and Wilson, A. M. "Group hunting within the Carnivora: physiological, cognitive and environmental influences on strategy and cooperation," *Behavioral Ecology and Sociobiology* Vol. 67, No. 1, 2013, pp. 1-17.

8. Stanford, C. B. "The influence of chimpanzee predation on group size and anti-predator behaviour in red colobus monkeys," *Animal Behaviour* Vol. 49, No. 3, 1995, pp. 577-587.

9. Scheel, D. "Profitability, encounter rates, and prey choice of African lions," *Behavioral Ecology* Vol. 4, No. 1, 1993, pp. 90-97.

10. Vaughn, R. L., Shelton, D. E., Timm, L. L., Watson, L. A., and Würsig, B. "Dusky dolphin (Lagenorhynchus obscurus) feeding tactics and multi-species associations," *New Zealand Journal of Marine and Freshwater Research* Vol. 41, No. 4, 2007, pp. 391-400.

11. Nøttestad, L., Fernö, A., and Axelsen, B. E. "Digging in the deep: killer whales' advanced hunting tactic," *Polar Biology* Vol. 25, No. 12, 2002, pp. 939-941.

12. Angelani, L. "Collective predation and escape strategies," *Physical review letters* Vol. 109, No. 11, 2012, p. 118104.

13. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. "Novel type of phase transition in a system of self-driven particles," *Physical review letters* Vol. 75, No. 6, 1995, p. 1226.

14. Lin, Y., and Abaid, N. "Collective behavior and predation success in a predator-prey model inspired by hunting bats," *Physical Review E* Vol. 88, No. 6, 2013, p. 062724.

15. Kamimura, A., and Ohira, T. "Group chase and escape," *New Journal of Physics* Vol. 12, No. 5, 2010, p. 053013.

16. Wang, X., He, M., and Kang, Y. "A computational predator–prey model, pursuit–evasion behavior based on different range of vision," *Physica A: Statistical Mechanics and its Applications* Vol. 391, No. 3, 2012, pp. 664-672.

17. Gaertner, U. "UAV swarm tactics: an agent-based simulation and Markov process analysis." NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF OPERATIONS RESEARCH, 2013.

18. Luke, S. "Multiagent simulation and the MASON library," *George Mason University* Vol. 1, 2011.

19. Strickland, L., Day, M. A., DeMarco, K., Squires, E., and Pippin, C. "Responding to unmanned aerial swarm saturation attacks with autonomous counter-swarms," *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*. Vol. 10635, International Society for Optics and Photonics, 2018, p. 106350Y.

20. Zhdankin, V., and Sprott, J. "Simple predator-prey swarming model," *Physical Review E* Vol. 82, No. 5, 2010, p. 056209.

21. Kolon, C., and Schwartz, I. B. "The Dynamics of Interacting Swarms," *arXiv preprint arXiv:1803.08817*, 2018.

22. D'Orsogna, M. R., Chuang, Y.-L., Bertozzi, A. L., and Chayes, L. S. "Self-propelled particles with soft-core interactions: patterns, stability, and collapse," *Physical review letters* Vol. 96, No. 10, 2006, p. 104302.

23. Gupta, S., and Baker, J. "Adversarial Swarms as Dynamical Systems," *arXiv preprint arXiv:2105.13294*, 2021.

24. Billings, S. A. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*: John Wiley & Sons, 2013.

25. Woolley, J. W., Agarwal, P., and Baker, J. "Modeling and prediction of chaotic systems with artificial neural networks," *International journal for numerical methods in fluids* Vol. 63, No. 8, 2010, pp. 989-1004.

26. Zhang, G. P. "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing* Vol. 50, 2003, pp. 159-175.

27. Teixeira, B. O. S., Castro, W. S., Teixeira, A. F., and Aguirre, L. A. "Data-driven soft sensor of downhole pressure for a gas-lift oil well," *Control Engineering Practice* Vol. 22, 2014, pp. 34-43.

28. Elman, J. L. "Finding structure in time," *Cognitive science* Vol. 14, No. 2, 1990, pp. 179-211.

29. Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., and Koumoutsakos, P. "Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics," *Neural Networks* Vol. 126, 2020, pp. 191-217.

30. Bishop, C. M. *Pattern recognition and machine learning*: springer, 2006.

31. Chatzis, S. P., and Demiris, Y. "Echo state Gaussian process," *IEEE Transactions on Neural Networks* Vol. 22, No. 9, 2011, pp. 1435-1445.

32. Vlachas, P. R., Byeon, W., Wan, Z. Y., Sapsis, T. P., and Koumoutsakos, P. "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* Vol. 474, No. 2213, 2018, p. 20170844.

33. Long, Y., She, X., and Mukhopadhyay, S. "HybridNet: integrating model-based and data-driven learning to predict evolution of dynamical systems," *arXiv preprint arXiv:1806.07439*, 2018.

34. Mackey, M. C., and Glass, L. "Oscillation and chaos in physiological control systems," *Science* Vol. 197, No. 4300, 1977, pp. 287-289.

35. Hochreiter, S. "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* Vol. 06, No. 02, 1998, pp. 107-116

36. Jaeger, H., and Haas, H. "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *science* Vol. 304, No. 5667, 2004, pp. 78-80.

37. Maass, W., Natschläger, T., and Markram, H. "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation* Vol. 14, No. 11, 2002, pp. 2531-2560.

38. Dominey, P. F. "Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning," *Biological cybernetics* Vol. 73, No. 3, 1995, pp. 265-274.

39. Dominey, P. F., Hoen, M., and Inui, T. "A neurolinguistic model of grammatical construction processing," *Journal of Cognitive Neuroscience* Vol. 18, No. 12, 2006, pp. 2088-2107.

40. Verstraeten, D., Schrauwen, B., d'Haene, M., and Stroobandt, D. "An experimental unification of reservoir computing methods," *Neural networks* Vol. 20, No. 3, 2007, pp. 391-403.

41. Schrauwen, B., Verstraeten, D., and Van Campenhout, J. "An overview of reservoir computing: theory, applications and implementations," *Proceedings of the 15th European Symposium on Artificial Neural Networks. p. 471-482 2007*. 2007, pp. 471-482.

42. Ferreira, A. A., Ludermir, T. B., and De Aquino, R. R. "An approach to reservoir computing design and training," *Expert systems with applications* Vol. 40, No. 10, 2013, pp. 4172-4182.

43. Skowronski, M. D., and Harris, J. G. "Automatic speech recognition using a predictive echo state network classifier," *Neural networks* Vol. 20, No. 3, 2007, pp. 414-423.

44. Antonik, P., Smerieri, A., Duport, F., Haelterman, M., and Massar, S. "FPGA implementation of reservoir computing with online learning," *24th Belgian-Dutch Conference on Machine Learning*. 2015.

45. Polydoros, A., Nalpantidis, L., and Krüger, V. "Advantages and limitations of reservoir computing on model learning for robot control," *IROS Workshop on Machine Learning in Planning and Control of Robot Motion, Hamburg, Germany*. 2015.

46. Lin, X., Yang, Z., and Song, Y. "Short-term stock price prediction based on echo state networks," *Expert systems with applications* Vol. 36, No. 3, 2009, pp. 7313-7317.

47. Zhang, C., Jiang, J., Qu, S.-X., and Lai, Y.-C. "Predicting phase and sensing phase coherence in chaotic systems with machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 8, 2020, p. 083114.

48. Antonelo, E. A., Camponogara, E., and Foss, B. "Echo State Networks for data-driven downhole pressure estimation in gas-lift oil wells," *Neural Networks* Vol. 85, 2017, pp. 106-117.

49. Maiorino, E., Bianchi, F. M., Livi, L., Rizzi, A., and Sadeghian, A. "Data-driven detrending of nonstationary fractal time series with echo state networks," *Information Sciences* Vol. 382, 2017, pp. 359-373.

50. Canaday, D., Griffith, A., and Gauthier, D. J. "Rapid time series prediction with a hardware-based reservoir computer," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 12, 2018, p. 123119.

51. Dale, M., Miller, J. F., Stepney, S., and Trefzer, M. A. "Evolving carbon nanotube reservoir computers," *International Conference on Unconventional Computation and Natural Computation*. Springer, 2016, pp. 49-61

52. Lu, Z., Pathak, J., Hunt, B., Girvan, M., Brockett, R., and Ott, E. "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 27, No. 4, 2017, p. 041102.

53. Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., and Ott, E. "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 27, No. 12, 2017, p. 121102.

54. Lu, Z., Hunt, B. R., and Ott, E. "Attractor reconstruction by machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 6, 2018, p. 061104.

55. Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Physical review letters* Vol. 120, No. 2, 2018, p. 024102.

56. Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B. R., Girvan, M., and Ott, E. "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 4, 2018, p. 041101.

57. Chattopadhyay, A., Hassanzadeh, P., Palem, K., and Subramanian, D. "Data-driven prediction of a multi-scale Lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ANN, and RNN-LSTM," *arXiv preprint arXiv:1906.08829*, 2019.

58. Zimmermann, R. S., and Parlitz, U. "Observing spatio-temporal dynamics of excitable media using reservoir computing," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 4, 2018, p. 043118.

59. Sapkota, P., Baker, J., and Gupta, S. "On the use of Reservoir Computing (Echo State Network) to predict the behaviour of an incompressible turbulent round jet," *Preprint*, 2020.

60. Krishnagopal, S., Girvan, M., Ott, E., and Hunt, B. R. "Separation of chaotic signals by reservoir computing," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 2, 2020, p. 023123.

61. Carroll, T. L. "Mutual Information and the Edge of Chaos in Reservoir Computers," *arXiv preprint arXiv:1906.03186*, 2019.

62. Mitchell, M., Hraber, P., and Crutchfield, J. P. "Revisiting the edge of chaos: Evolving cellular automata to perform computations," *arXiv preprint adap-org/9303003*, 1993.

63. Crutchfield, J. P. "Between order and chaos," *Nature Physics* Vol. 8, No. 1, 2012, pp. 17-24.

64. Carroll, T. L. "Dimension of reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 1, 2020, p. 013102.

65. Carroll, T. L., and Pecora, L. M. "Network structure effects in reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 29, No. 8, 2019, p. 083130.

66. Zhang, C., Jiang, J., Qu, S.-X., and Lai, Y.-C. "Predicting phase and sensing phase coherence in chaotic systems with machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 8, 2020, p. 083114.

67. Hinaut, X. "neuronalX." 2020,Github,last accessed on July 31st 2021

68. Trouvain, N., Pedrelli, L., Dinh, T. T., and Hinaut, X. "ReservoirPy: an Efficient and User-Friendly Library to Design Echo State Networks," *International Conference on Artificial Neural Networks*. Springer, 2020, pp. 494-505

69. Gould, H., and Tobochnik, J. *An Introduction to Computer Simulation Methods: Applications to Physical Systems*: Addison-Wesley, 1996.

70. Chen, Y., and Yang, H. "Multiscale recurrence analysis of long-term nonlinear and nonstationary time series," *Chaos, Solitons & Fractals* Vol. 45, No. 7, 2012, pp. 978-987

71. Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A. "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena* Vol. 16, No. 3, 1985, pp. 285-317.

72. Grebogi, C., Ott, E., and Yorke, J. A. "Crises, sudden changes in chaotic attractors, and transient chaos," *Physica D: Nonlinear Phenomena* Vol. 7, No. 1-3, 1983, pp. 181-200.

73. Ahmed, M. U., and Mandic, D. P. "Multivariate multiscale entropy: A tool for complexity analysis of multichannel data," *Physical Review E* Vol. 84, No. 6, 2011, p. 061918.

74. Costa, M., Goldberger, A. L., and Peng, C.-K. "Multiscale entropy analysis of biological signals," *Physical review E* Vol. 71, No. 2, 2005, p. 021906.

75. Lukoševičius, M., Jaeger, H., and Schrauwen, B. "Reservoir computing trends," *KI-Künstliche Intelligenz* Vol. 26, No. 4, 2012, pp. 365-371.

76. Lukoševičius, M. "A practical guide to applying echo state networks," *Neural networks: Tricks of the trade*. Springer, 2012, pp. 659-686.

77. Shofer, N. "Reservoir Computing: Memory, Nonlinearity, and Spatial Observers." Reed College, 2018.

78. Hoerl, A. E., and Kennard, R. W. "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics* Vol. 12, No. 1, 1970, pp. 55-67.

79. Gupta, S., and Baker, J. "Adversarial Swarms as Dynamical System," *Preprint*, 2020.

80. Lorenz, E. N. "Deterministic nonperiodic flow," *Journal of atmospheric sciences* Vol. 20, No. 2, 1963, pp. 130-141.

CHAPTER 4

EFFECT OF INDIVIDUALITY ON ADVERSARIAL SWARM BEHAVIOUR: A HYBRID PARALLEL ECHO STATE NETWORKS APPROACH

4.1 Abstract

In the current work, a massively parallel Echo State Network was developed to predict the dynamics of an Adversarial Swarm system. The Adversarial Swarm system consisted of two types of agents-the Attackers and the Defenders who had conflicting objectives in 2D Euclidean space. Each instance of ESN in the massively parallel framework was trained n individual spatio-temporal data of every agent. The large-scale ESN was successfully able to capture the dynamics of the interacting adversarial swarm system. The hyperparameters for each ESN were evaluated using a uniform grid search. The optimal hyperparameter obtained for every individual agent showed considerable variance that indicated that each ESN was qualitatively different from one another, which further implied that every agent the Adversarial swarm reacted uniquely when a uniform stimulus was applied across them.

4.2 Introduction

Swarms are fundamentally complicated systems with non-linear dynamics that a physics-based model may accurately represent. Independent, autonomous agents are the constituents of a swarm that is widespread in nature from ranging from ant colonies [1], flocks of birds [2], and schools of fishes[3]. Swarms form the basis of extensive modern-day engineering applications ranging from spacecraft [4], UAVs [5], robots [6], and optimization algorithms[1]. Simulating natural swarm behavior for engineering applications is fundamentally problematic since natural

systems have intrinsic flexibility and scalability, sometimes impossible to achieve with digital computers.

In nature, there are two forms of swarm interactions: adversarial and symbiotic, in which swarms compete or collaborate for resource utilization. Symbiotic swarms found in the animal kingdom include multi-species group [7](Bailey et al., 2013)hunting, were in different groups of species team up (cooperation) with each for hunting groups of prey. Adversarial swarms are abundant in natural environments, such as predators engaging with prey groups, and can be found both in aquatic and terrestrial environments. Most of such interactions take place for foraging purposes. Natural examples of Adversarial Swarms include groups of omnivorous Chimpanzees hunting groups of Red Colobus Monkeys[8] and groups of predator Lions hunting herds of Zebras[9].In the aquatic environment, a multi-species association of Dolphins with Seals and Dogfish for feeding schools of small fish[10], groups of Killer Whales, and a large number of Herring, where the former would force the later to dive up by almost 150 meters[11], which would enable more effective foraging.

Gupta et al.[12], developed a physics-based dynamical Adversarial Swarm model with well-defined intra-swarm and inter-swarm forces. In unbounded 2D Euclidean Space, the model consists of two separate interacting swarms: the Attackers and the Defenders, who have opposing agendas. The Defenders protect the ‘Goal’ a point of interest in unbounded 2D Euclidian space. In contrast, the Attacker's main objective is to intercept the Goal while continually evading the Defenders. The Defenders' swarms protect the Goal by swarming around it and blocking any Attackers agent trying to reach the Goal. If an Attacker and Defender agents are very close to each other, if the distance between them is less than a predefined criterion, they were assumed to have collided and were consequently arrested for further participation. The arrested agents

were inactive for the remainder of the simulation. The simulation was assumed to have a binary outcome or a final state, wherein either the Attacker or the Defenders emerged as the dominant swarm. The Attacker swarm was considered dominant if an agent in the swarm successfully intercepted the Goal during the simulation. If there were no remaining agents in the Defender swarm at any point in the simulation, the Attackers were considered dominant. If the Defenders successfully defended the Goal before the end of the simulation or if no Attackers were left in the simulation, the Defenders were considered as the dominant swarm. If no agents were left in either of the swarms (i,e the agents compromise each other off in the engagement), the Defenders were considered dominant in the simulation as the Goal was successfully protected from the predation of the Attacker swarm. The scenarios mentioned above formed the basis of 'Simulation Ending Criteria' presented in[13] . The non-linear time-series data obtained from the simulations performed in[13] were studied using various tools that included time-series plots, recurrence plots, attractor plots, and the Largest Lyapunov Exponent (LLE). The system was strongly investigated for the presence of Chaos. As a vital parameter of the system, the number of Attackers and the Defender agents making up each swarm is varied to study the simulation's final state. The Largest Lyapunov Exponent for each case is also evaluated to probe the presence of Chaos. The degree of determinism in the data or the complexity of the system was found by assessing the Multiscale Entropy of the non-linear time series data.

The current state of the art reveals that ESNs have been very successful in predicting chaotic dynamic systems. ESNs in the contemporary literature have been found to use three configurations while predicting chaotic behaviors, namely, observer mode (non-autonomous or predictive mode), where model free prediction is achieved by utilizing limited state variables [14], the generative (or autonomous) mode in which during the prediction the output of a previous

time-step is fed as the input in the Reservoir [15-18] and finally custom ensemble methods where ESNs are used in conjunction with knowledge-based models[17]. The authors of the previously mentioned works successfully verified and validated these techniques in the simple Lorentz system[19] , Lorentz-96 system[20], Kuramoto-Sivashinsky (KS) system[16, 18], the Rössler system[18], and dynamics of excitable media such as the Barkley model and the Bueno-Orovio-Cherry-Fenton model[21]. ESNs have also successfully predicted large-scale dynamical systems such as Large Eddy simulation of an incompressible turbulent round jet by implementing massively large-scale parallel reservoirs. Hardware-based reservoir computers have also successfully predicted dynamical systems[22] , such as the Mackey-Glass system. Krishnagopal [23] studied the effectiveness of a reservoir computer to separate chaotic signals and concluded that their results were better than the Wiener filter obtained from the same training data. Several studies have also been conducted to have a holistic understanding of a Reservoir computer's inherent dynamics, which would enable its effective use while predicting dynamic systems. Carroll[24] used an RC-ESN at the 'Edge of Chaos'[25, 26] region to perform predictions and concluded that it does not necessarily improve the performance. Carroll [27] also conducted studies on the dimension of Reservoir computers and concluded the increase of fraction dimension occurs inside the Reservoir with the increase of its dimension, which may adversely affect the performance of Reservoir Computer, Carroll [28] also conducted studies on the network structure of the RC-ESN. Zhang [29] conducted studies in the sensory phase coherence of two parallel reservoirs and concluded that short-term prediction is possible, but parallel reservoirs are limited in sensing the collective dynamics of a coupled chaotic dynamic of the entire system in the long run. Chang [30] used Reservoir Computing to study temperature fluctuations in a Rayleigh-Bernard convection problem. Gupta et al,[12] used two parallel hybrid

ESNs to predict the group dynamics of Adversarial Swarms[13], where each ESN was assigned to Attackers and Defenders, respectively.

An ESN capable of predicting spatio-temporal systems with high accuracy must have perfectly hyperparameters, including spectral radius scaling factor, leaking rate, probability of non-zero connections, and regularization coefficients. This task can be achieved by performing a uniform or random grid search that is essential to produce a well-trained network capable of predicting the highly transient dynamical systems. A large-scale parallel ESN setup was developed in the current work wherein individual reservoirs will be assigned to each agent of the Attackers and the Defenders, respectively. The respective agents were trained separately, and a comprehensive hyperparameter random search was performed. Random grid search was preferred over a comprehensive grid search as owing to limitations of computing resources. Also, it was found in the literature that the former does not always yield the best possible results [31].

To date, research has focused on applying a single simple rule set to encode the response of all agents in each swarm to a specific stimulus. Intuitively, it is known that all organisms (even those of the same species) do not react to identical stimuli uniformly. By using a step-by-step approach, the variations in individual responses affecting the behavior of a given swarm will be explored, in addition to its effect on the interaction between two adversarial swarms. The following are the main intellectual contributions of this paper:

- Examine how the dynamics of the two Adversarial Swarms will change when each agent is represented as an individual by a unique Echo State Network, trained using data specific to a given agent.
- Uniform grid searches to obtain the optimal hyperparameter of each ESN corresponding to each agent in the massively parallel ESN framework.

- Statistical analysis of hyperparameter data for each hybrid ESN assigned to each ESN to explore the idea of Individuality in agents.

This paper is organized as follows- section 4.3 discusses the methods and formulations, which contains the Adversarial Swarm model formulations and the massively parallel ESN configurations. Section 4.4 includes the verification and validation of the ESN. Section 4.5 covers the results and discussion, followed by the conclusion in section 4.6.

4.3. Formulations

4.3.1 Adversarial Swarm Model

A physics-based agent-based model is developed to study the dynamics of two interacting Adversarial Swarms: The Attacker Swarm and the Defender Swarm (hence, referred to as 'Attackers' and 'Defenders' respectively). The agents have conflicting objectives; the Defenders protect a point of interest in unbounded 2D Euclidean space by swarming around the Goal along a sphere of influence. In contrast, the Attackers' main task is to intercept the Goal while constantly evading the Defenders. They actively chase the former in a perimeter around the Goal or a sphere of influence. The individual swarms in the swarm system are modeled based on a Lagrangian-based approach having primarily two types of forces- 'inter' and 'intra' swarm forces; the inter-forces are used to model the interaction between the agents of the Adversarial Swarms, respectively. The intra-forces are used to model the forces between members of the same swarm. Each swarm can be generalized as a collection of N agents in a 2-Dimensional space with position and velocity vectors. The governing equation describing the dynamics of the two interacting swarms- the Attackers and the Defenders are derived based on Newton's second law of motion and are given by the following equations[13].

$$\ddot{\vec{X}}_{A,i} = \dot{\vec{V}}_{A,i} = \frac{1}{m_{A,i}}\left(\sum_{\substack{Ai=1 \\ Ai\neq Ak}}^{N_A} -(\nabla\varphi_{A,ik}) + \sum_{Ai=1}^{N_D} -\nabla(k_{rep}r_{ij}^{-1}) - \nabla(-k_{obj}r_{iG}^{2}) + \right.$$

$$\left.(\alpha_A - \beta_A|\vec{V}_{A,i}|^2)\vec{V}_{A,i}\right. \tag{4.1}$$

$$\dot{\vec{X}}_{A,i} = \vec{V}_{A,i} \tag{4.2}$$

$$\ddot{\vec{X}}_{D,j} = \dot{\vec{V}}_{D,j} = \frac{1}{m_{D,i}}\left(\sum_{\substack{Dj=1 \\ Dj\neq Dh}}^{N_D} -(\nabla\varphi_{D,jh}) + \sum_{Dj=1}^{N_A} -\nabla(-k_{att}r_{ji}^{-1}) - \nabla(\varphi_{jG}) + (\alpha_D - \right.$$

$$\left.\beta_D|\vec{V}_{D,j}|^2)\vec{V}_{D,j}\right. \tag{4.3}$$

$$\dot{\vec{X}}_{D,j} = \vec{V}_{D,i} \tag{4.4}$$

Eqns. (4.1-4.4) are the principal equations for the Adversarial Swarm model that is subject to given initial conditions $\vec{V}_{A,i}(t=0), X_{A,i}(t=0), \vec{V}_{D,i}(t=0), X_{D,i}(t=0)$ of individual agents in the respective swarms are known. Eqs. (4.1)-(4.4) are numerically integrated using a customized fixed time-step 4th order Runge-Kutta explicit solver[32].

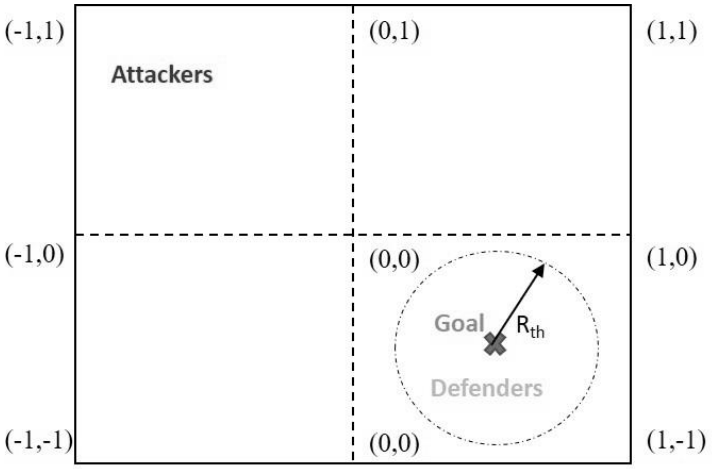

Figure 4.1: Illustration of the domain for Adversarial Swarm Simulation.

Multiscale Entropy(MSE)[33, 34] was also evaluated from the center of mass time-series to study the complexity of the interacting swarms. The MSE results (Fig. 4.2) revealed a greater degree of randomness for the Defenders compared to Attackers due to the nature of the former's

role in the simulation. Overall, the maximum value of MSE obtained was ~1.40 for both the Attackers and Defenders combined. The average MSE for the Attackers and Defenders is 1.04 and 1.25, respectively when calculated over time scales from 1 to 20. As a result of the medium degree of randomness of the non-linear center of mass time-series for both classes of agents, the prediction of spatio-temporal data was reasonable.

Table 4.1: Case matrix for Adversarial Swarm simulation

Case	N_A	N_D	N_A/N_D	Max Sim Time
A	2	18	0.11	100
B	5	15	0.33	100
C	7	13	0.53	100
D	10	10	1	100
E	13	7	1.85	100
F	15	5	3.0	100
G	18	2	9.0	100

4.3.1: Echo State Networks Preliminaries

An input vector u(t) with K units was fed into a dynamic 'Reservoir' with N units in a generic Echo State Network (ESN). The Reservoir was connected to an output layer y(t)with L units. At discrete time t, the Reservoir receives input, which is subsequently coupled with the reservoir state to produce output at $t+\partial t$ .In Echo State Networks, an RNN with leaky-integrated discrete-time continuous value units is employed. The following are the ESN equations:

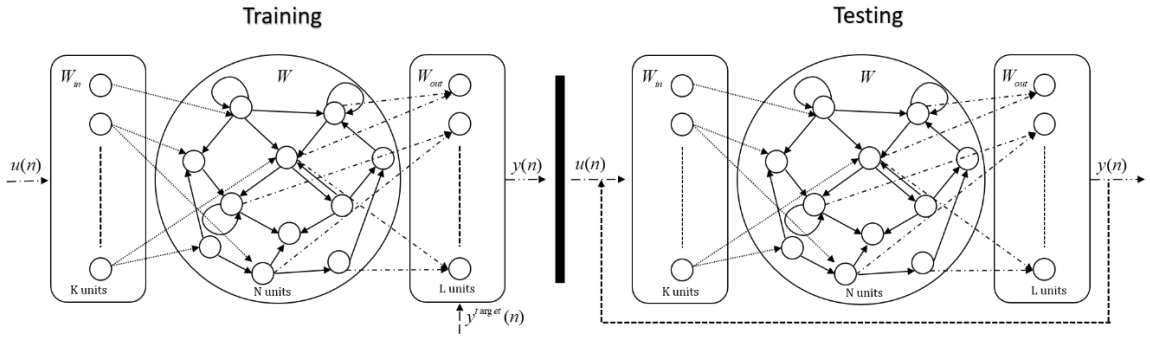

Figure 4.2: Training and Testing of a generic Echo State Network.

Echo State Networks typically use an RNN with leaky-integrated discrete-time continuous value units[35](Lukoševičius et al., 2012). The equations of update used in this study are obtained from[35] and are given by the following equations:

$$\tilde{x}(n) = \tanh(W^{in}[1; u(n)] + Wx(n-1)) \tag{4.5}$$

$$x(n) = (1-\alpha)x(n-1) + \alpha\tilde{x}(n) \tag{4.6}$$

Where $x(n) \in \mathbb{R}^{N_x}$ represents neuron activations of the Reservoir and $\tilde{x}(n) \in \mathbb{R}^{N_x}$ is its update at every time step n. The tanh() function is used as the sigmoid wrapper for the neurons. $\alpha \in (0,1]$ was the leaking rate, [;] represents a vertical vector (or matrix) concatenation $W^{in} \in \mathbb{R}^{N_X \times (1+N_U)}$ and $W^{N_X \times N_X}$ is the input weight matrix and recurrent weight matrix, respectively[35, 36]. Once the Reservoir is trained, the output weight matrix is generated using linear regression. The linear readout layer is defined as,

$$y(n) = W^{out}[1; u(n); x(n)] \tag{4.7}$$

Where $y(n) \in \mathbb{R}^{N_y}$ represents the network output $W^{out} \in \mathbb{R}^{N_y \times (1+N_u+N_X)}$ is the output weight matrix, and [;] represents a vertical vector concatenation. The network output weight matrix is learned by comparing the network output y(n) with the target output $y^{target}(n)$. This task is achieved by minimizing the Root Mean Square Error (RMSE) between the network output and the target output [35, 37].

$$E(y, y^{target}) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{T}\sum_{n=1}^{T}(y_i(n) - y_i^{target}(n))^2} \tag{4.8}$$

The operation of the ESN, as defined by[38], can be briefly summarized by the following steps:

A. Generation of a random reservoir with a given input weight matrix W^{in}, recurrent weight matrix W, and a given leaking rate α.

B. The training input u(n) drives the network, and reservoir activation units x(n) are collected.

C. A linear readout layer obtains the network output. The output weights W_{out} are computed by minimizing the error between the network output $y(n)$ and the target output $y_i^{target}(n)$.

D. Finally, compute output in the prediction phase using the trained network on the recycled data (i, e, the data obtained in the previous time-step) in case of generative mode or a limited component time-series in predictive mode.

A ESN capable of forecasting with high accuracy can be produced by effectively tuning its critical hyperparameters. This task was achieved by performing a comprehensive uniform grid search essential to producing a well-trained network capable of predicting the highly transient dynamical behavior of the Adversarial Swarm system. The critical hyperparameters of the ESN - spectral radius scaling factor, leaking rate, probability of non-zero connections, and regularization coefficients were chosen for the grid search.

The readouts from the ESN in Eqn. (4.8) can be rewritten using matrix notation as:

$$Y = W^{out}X \tag{4.9}$$

Where $Y \in \mathbb{R}^{N_Y \times T}$ are is $y(n)$ and $X \in \mathbb{R}^{(1+N_U+N_X)\times T}$ are all $[1; u(n); x(n)]$ produced by presenting the Reservoir $u(n)$, both collected into respective matrices by concatenating the column-vectors horizontally over the training period n=1,……, T. For notational simplicity, X is used instead of $[1;\ U;\ X]$. The optimized weights W^{out} minimize the RMSE between $y(n)$ and $y^{target}(n)$ are obtained by solving the following overdetermined linear equations.

$$Y^{target} = W^{out}X \tag{4.10}$$

The output weights W^{out} in equation (11) are found out by minimizing the RMSE in

Eqn. (8) by the following equation:

$$W^{out} = \text{argmin}\frac{1}{N_y}\sum_{i=1}^{N_y}(\sum_{i=1}^{T}(y_i(n) - y_i^{target}(n))^2 + \beta||W_i^{out}||^2) \quad (4.11)$$

4.3.3: Hybrid Echo State Networks (ESN): Massively Parallel Configuration

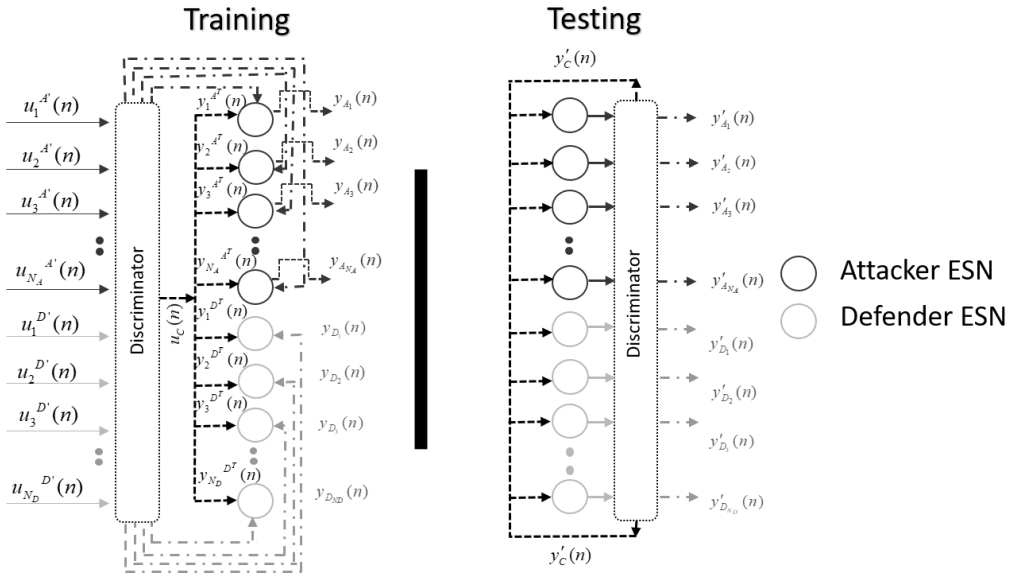

Figure 4.3: Massively parallel hybrid ESN setup

A massively parallel hybrid ESN architecture was operated to operate in the autonomous (or generative) mode. The position vector for each individual agent[13] was used to train a separate instance of ESN, which ran parallelly using multiple cores using a mpi4py (Message Passing Interface) architecture in python[39] . The purpose of the 'Discriminator' is to preprocess the data during the training phase and to post-process the data at the end of every respective time-step to enforce the simulation ending criteria, check agent compromise criteria, and in case of any agent collision post-process, the data further for input into the massively parallel hybrid ESN for the prediction of the subsequent time step.

Each ESN was provided with a standard input consisting of the position vector data for all the agents taken together in this massively parallel ESN configuration. In contrast, the target data used during the training phase corresponded to that of individual agents. This novel training

and testing scheme enabled the ESN to learn the interaction of a particular agent to the agents of its own swarm as well as to the agent of its competing counterpart, which is also the case in the Adversarial Swarm simulation performed in [13] where all the agents were globally coupled. During the prediction (or testing) phase, the predicted data of all the individual agents of the swarm in the preceding time step was passed as the driving signal to generate a future prediction. The training of the reservoirs used in this setup was carried out independently.

A comprehensive grid search was performed to tune hyperparameters of the hybrid ESN. It was hypothesized that even though the Adversarial system in the author's model [13] consisted of homogenous agents, each agent would respond differently when a uniform stimulus was applied. In the simulation environment, all the agents in the respective swarm were assumed to be globally coupled both intrinsically (within the members of the same swarm) and extrinsically (within the members of the competing swarm), thus making their reaction to any engagement or stimuli nonuniform. It is hypothesized that, in phase 3, the hyperparameters for every respective trained ESN would be unique, further strengthening the idea of the Individuality of every autonomous agent in a complex adaptive system. A detailed statistical analysis of the hyperparameter data obtained from all the independently trained parallel ESNs was carried out to investigate the central hypothesis of the final phase of this investigation.

4.4. Results and Discussion

4.4.1 Prediction of the Dynamics of Adversarial Swarms

In the current study, 7 cases were simulated (as described in table 2), and the corresponding spatiotemporal data for every individual agent was used to train and test the massively parallel echo state network as described in the preceding sections. A comprehensive hyperparameter search was conducted for each individual agent. The primary hyperparameters

considered for the grid search were – number of reservoir nodes, leak rate, and the probability of non-zero connections; another hyperparameter-the regularization coefficient was dropped for this study as it was found out from several numerical experiments that it does not have a substantial contribution towards the hybrid ESN's performance.

Table 4.2: Prediction outcome of the test cases outlined in Table 4.1

Case	N_A	N_D	Average NRMSE	% Time Difference	Predicted End	Actual Simulation End
A	2	18	0.086	0.0	Defenders Won	Defenders Won
B	5	15	0.017	0.0	Defenders Won	Defenders Won
C	7	13	0.279	0.0	Defenders Won	Defenders Won
D	10	10	0.234	0.0	Defenders Won	Defenders Won
E	13	7	0.253	0.0	Attackers Won	Attackers Won
F	15	5	0.068	0.0	Attackers Won	Attackers Won
G	18	2	0.016	0.0	Attackers Won	Attackers Won

The hyperparameter search was achieved by first generating all the possible unique combinations of the hyperparameters for all the agents combined. The massively parallel hybrid ESN framework was then trained and subsequently used for the prediction task for each combination respectively. The hyperparameter combination with the least NRMSE and the least percentage error in total time was the best solution. As determined from earlier numerical experiments in Gupta et al. (Soham Gupta, 2021), 44% of the available time-series data of every individual agent in both swarms respectively were used to train each instance of ESN in a massively parallel setup. The center of mass time-series was considered the comparison metric as it can demonstrate an overall interaction of the swarm interaction. The NRMSE for each case is described in Table 4.2. The following paragraphs discuss the results of each case in detail.

In the case of A, the number of Defenders was 18 compared to just 2, making up the Attacker swarm. It was observed that the Defenders were easily able to compromise the Attacker

agents quite early in the simulation; the massively parallel hybrid ESN predicted the same consequence, the NMSE obtained was 0.086 for the center of mass time-series plots as visualized in Fig 4.4. The outcome of the simulation and the prediction were the same; the Defenders were easily able to emerge as the dominant party in the simulation.

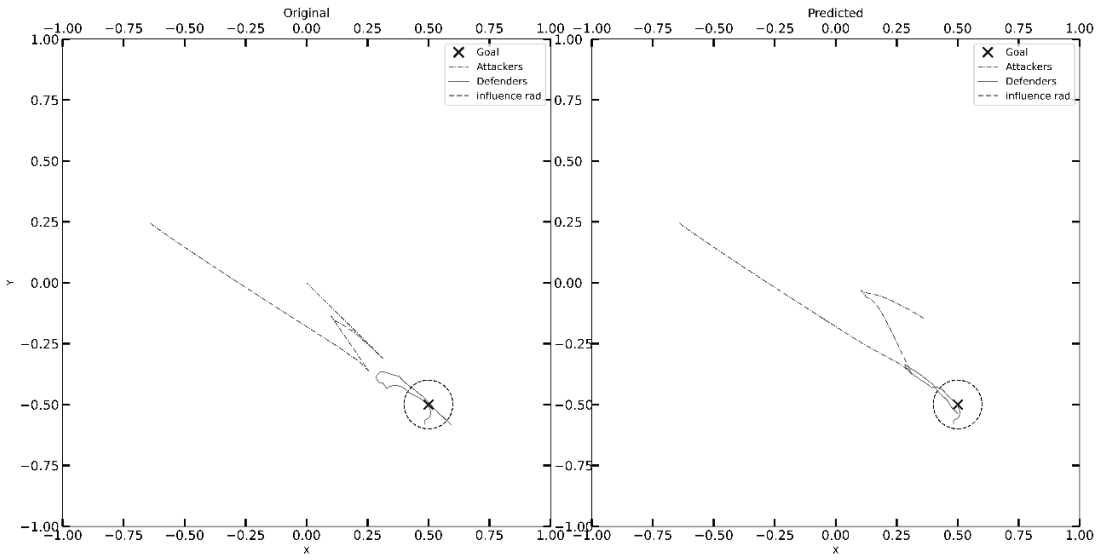

Figure 4.4: Swarm interaction trajectories for the center of mass time-series (both original and predicted) for Case A.

In case B (Fig.4.5), similar behavior was observed as the Defenders emerged as the dominant swarm, easily compromising the agents of the attacker swarm. The prediction from the ESN agreed with the original simulation; the NRMSE for this was 0.017 (Table 4.2).

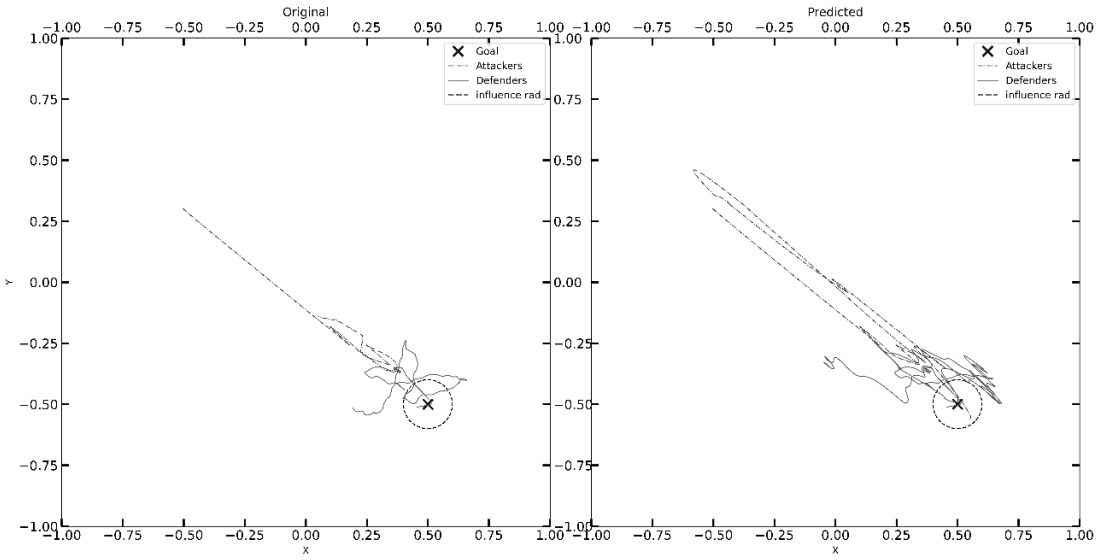

Figure 4.5: Swarm interaction trajectories for the center of mass time-series (both original and predicted) for Case B.

In case C (Fig. 4.6), the Defenders also emerged as the victorious swarm easily comprising the Attacker agents. The Defenders were able to do so by protecting the Goal from the interception of the Attacker agents until the end by also compromising the former in the process. Fig.4.6 revealed the ESN was able to capture the dynamics of the Defenders to a greater extent than the Attacker agents; however, it must be noted that the overall ergodic properties of the system remained the same, and the outcome of the simulation was predicted correctly by the ESN.

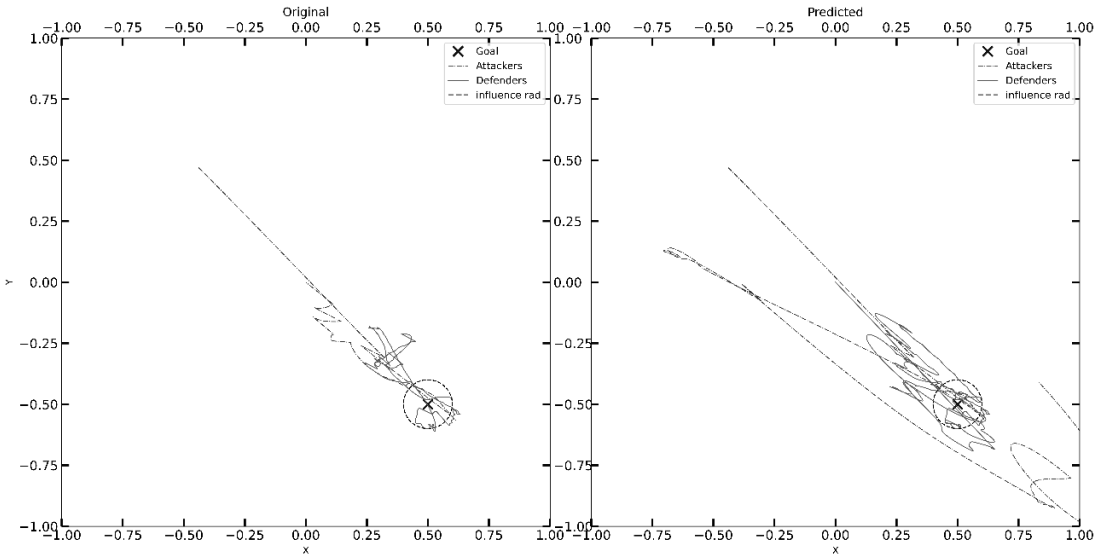

Figure 4.6: Swarm interaction trajectories for the center of mass time-series (both original and predicted) for Case C.

In case D (Fig. 4.7), it was observed that the Defenders were able to again emerge as the dominant party in the simulation by compromising the Attacker agents in action. The ESN successfully captured the dynamics of the overall interaction as seen from the center of mass time-series plot for Case D.

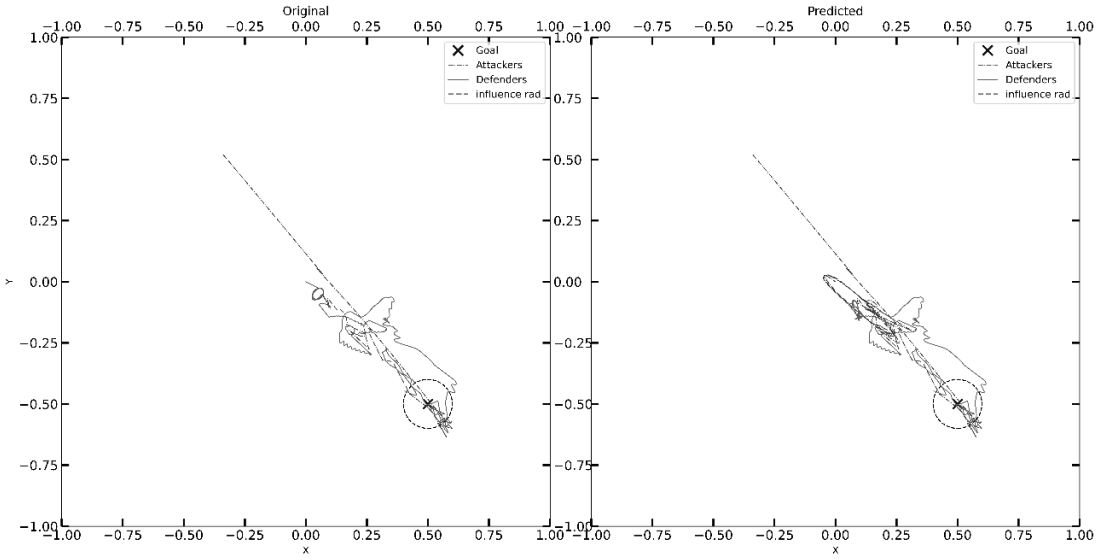

Figure 4.7: Swarm interaction trajectories for the center of mass time-series (both original and predicted) for Case D.

In case E (Fig. 4.8), the Attackers emerged as the victorious swarm by compromising the Defender agents early in the simulation. The ESN was also able to predict the outcome correctly, as it captured the overall dynamics of the interacting swarms.

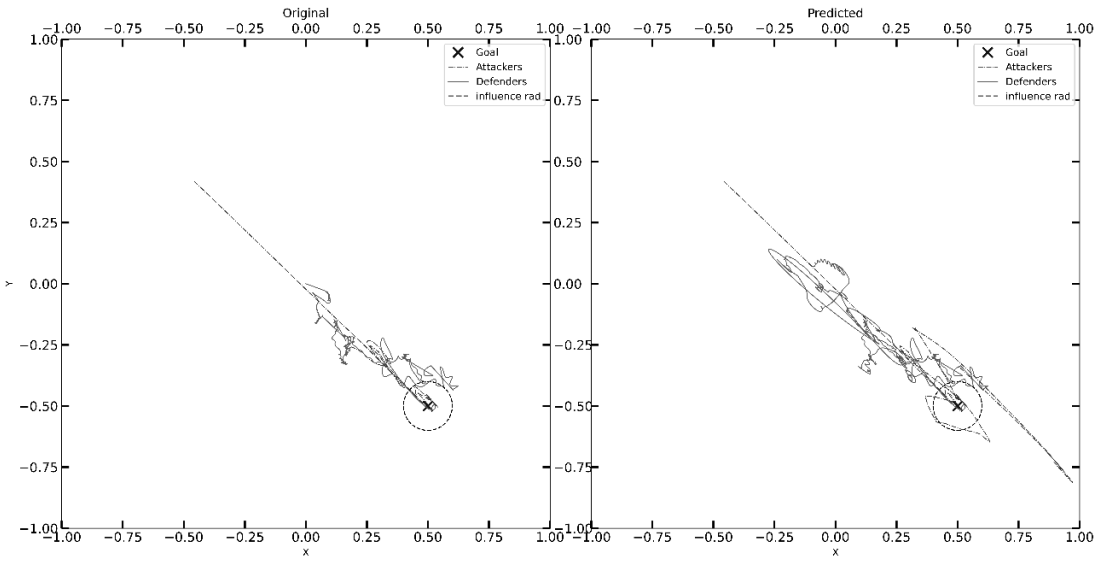

Figure 4.8: Swarm interaction trajectories for the center of mass time-series (both original and predicted) for Case E.

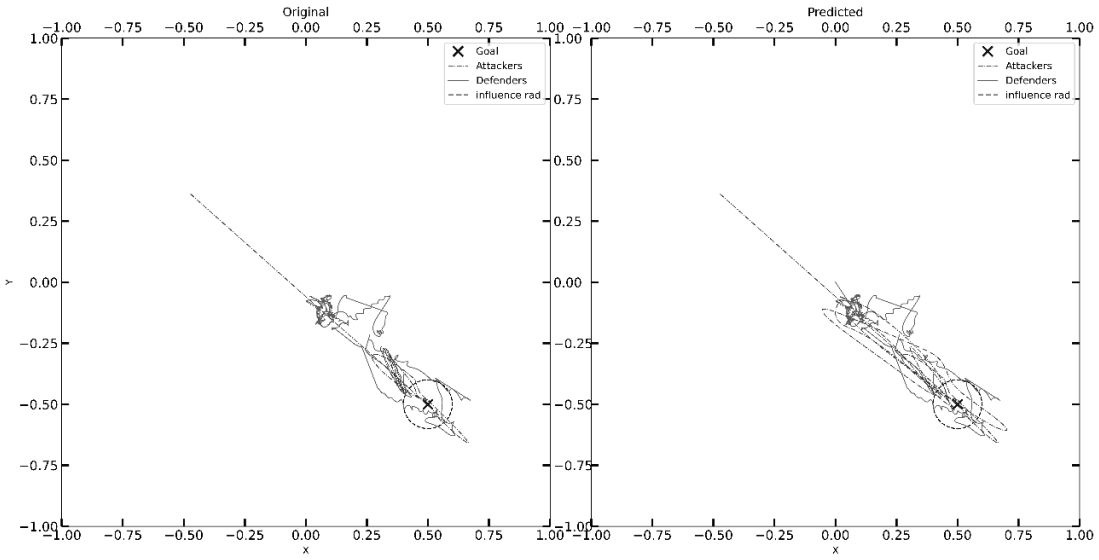

Figure 4.9: Swarm interaction trajectories for the center of mass time-series (both original and predicted) for Case F.

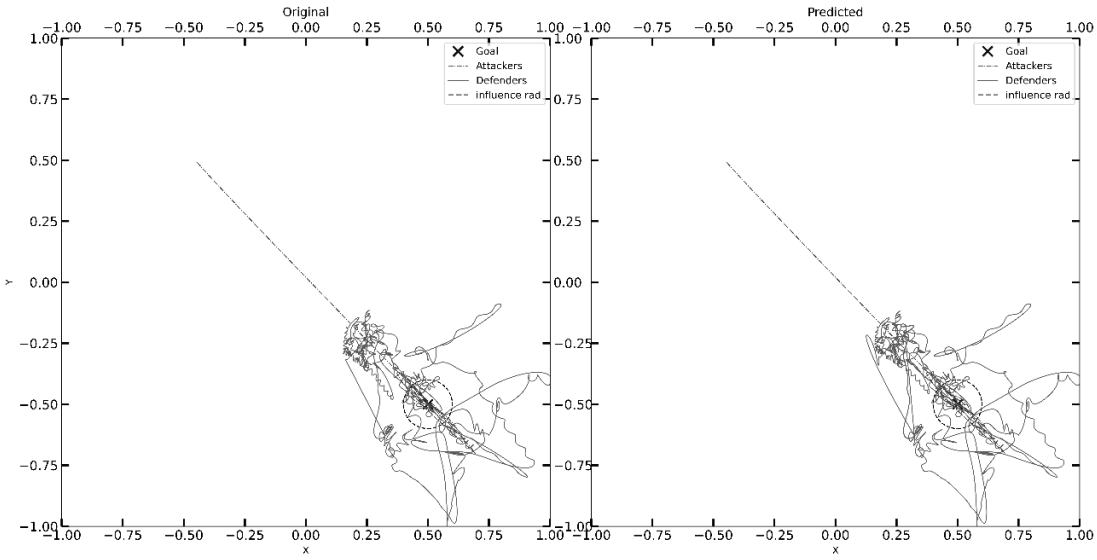

Figure 4.10: Swarm interaction trajectories for the center of mass time-series (both original and predicted) for Case G.

In case G and case F (Figs 4.9, Fig 4.10), the Attacker emerged as the victorious swarm by compromising the Defender agents in action. The novel was able to predict the outcome of the simulation successfully. The NRME(s) for these cases are 0.068 and 0.016, respectively, indicating the prediction was of good quality.

4.4.2 Uniform Hyperparameter search and Individuality of agents

A comprehensive uniform hyperparameter search was conducted for the significant hyperparameters of the massively parallel hybrid ESN-number of reservoir nodes, leak rate, and probability of non-zero connections. These hyperparameters were important as they affected the ESN ability to map a high dimensional dynamical system, the information flow inside an ESN, and the ability to forget past information to predict the future. The minimum, maximum, and step size of the significant hyperparameters used in the uniform search can be found in Table 4.3.

Table 4.3: Significant Hyperparameter Ranges used in the Uniform Grid search

Hyperparameter	Min	Max	Step
Number of Reservoir Nodes	300	450	5
Leak Rate	0.05	0.20	0.02
Probability of non-zero connections	0.05	0.20	0.02

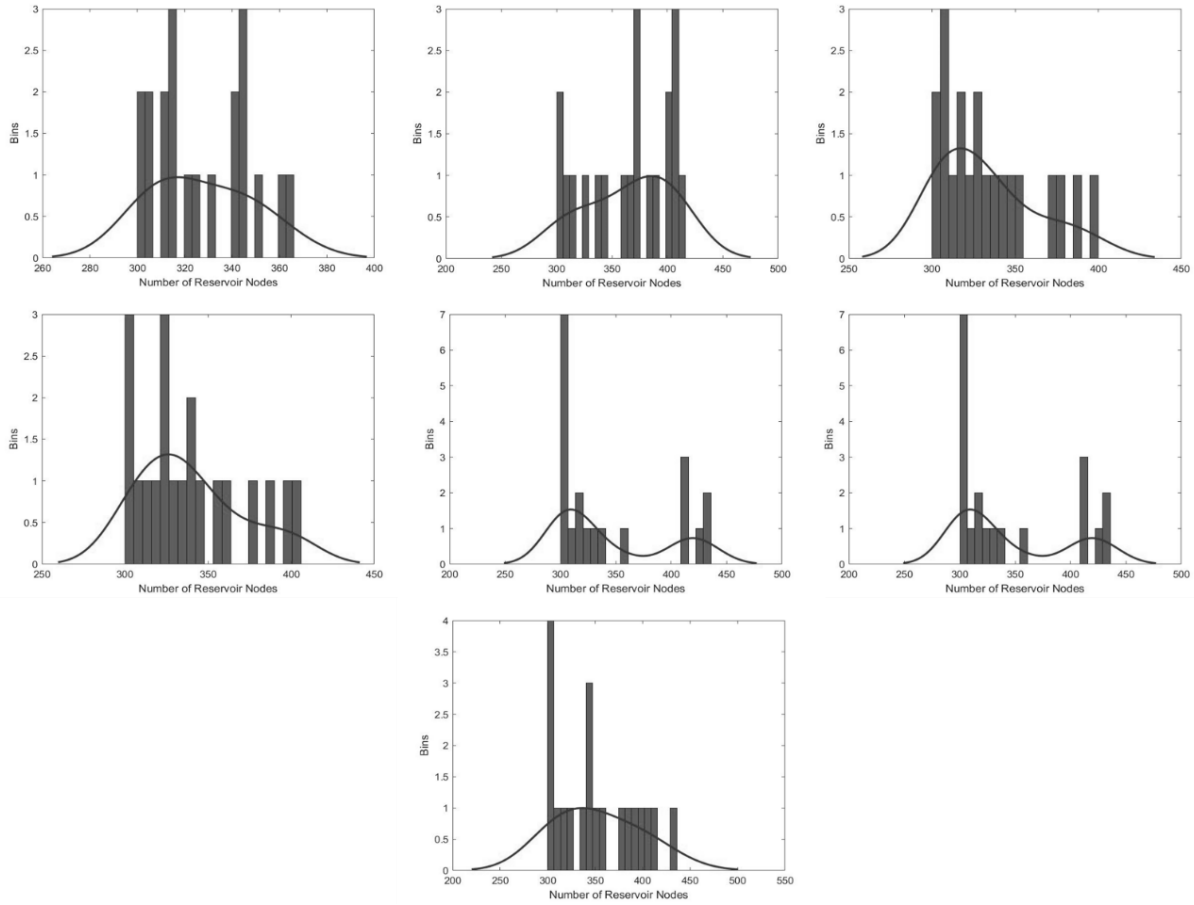

Figure 4.11: Histogram and probability distribution plot for Number of Reservoir Nodes (Hyperparameter) for Cases A through G for all Attackers and Defenders combined. Cases A through G are arranged in a row-wise sequence.

Table 4.4: Significant Hyperparameter Statistics for all cases taken together

Hyperparameter	Mean	Median	Standard Deviation
Number of Reservoir Nodes	341.61	337.5	34.16
Leak Rate	0.107667	0.11	0.040185
Probability of non-zero connections	0.088	0.07	0.03

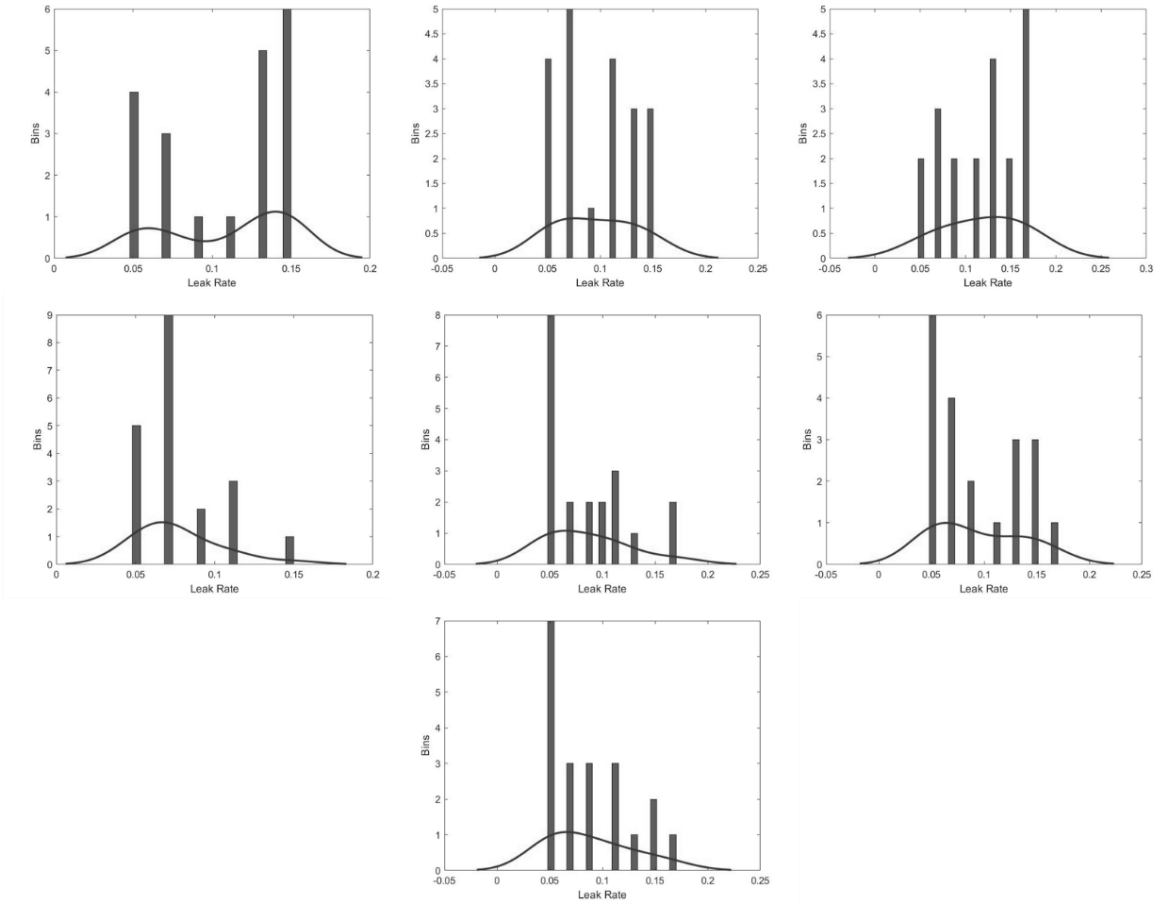

Figure 4.12: Histogram and probability distribution plot for Leak Rate (Hyperparameter) for Cases A through G for all Attackers and Defenders combined. Cases A through G are arranged in a row-wise sequence.

A comprehensive uniform gird search was conducted for the significant hyperparameters of each ESN of the massively parallel hybrid model revealed that the hyperparameters were not the same for all the agents. However, they were the same for some. The spatio-temporal data coming from every individual agent might be qualitatively the same; however, this is hardly the

case in real-world complex (natural and engineered) systems. Every agent reacts in a swarm reacts slightly differently when a uniform stimulus is applied. Thus, the presence of heterogeneity in homogenous agents was observed; the ESN ideally should have had the same hyperparameters as the agents making up the Adversarial Swarm, as these agents were fundamentally similar, having conflicting objectives. Thus, the ESN needed to be trained for each agent separately so as have the best possible prediction. The massively parallel ESN setup captured the Individuality of every agent, whereas the two parallel hybrid ESNs working in parallel would have fallen short. This behavior was observed from the histograms of the hyperparameters- Reservoir Node

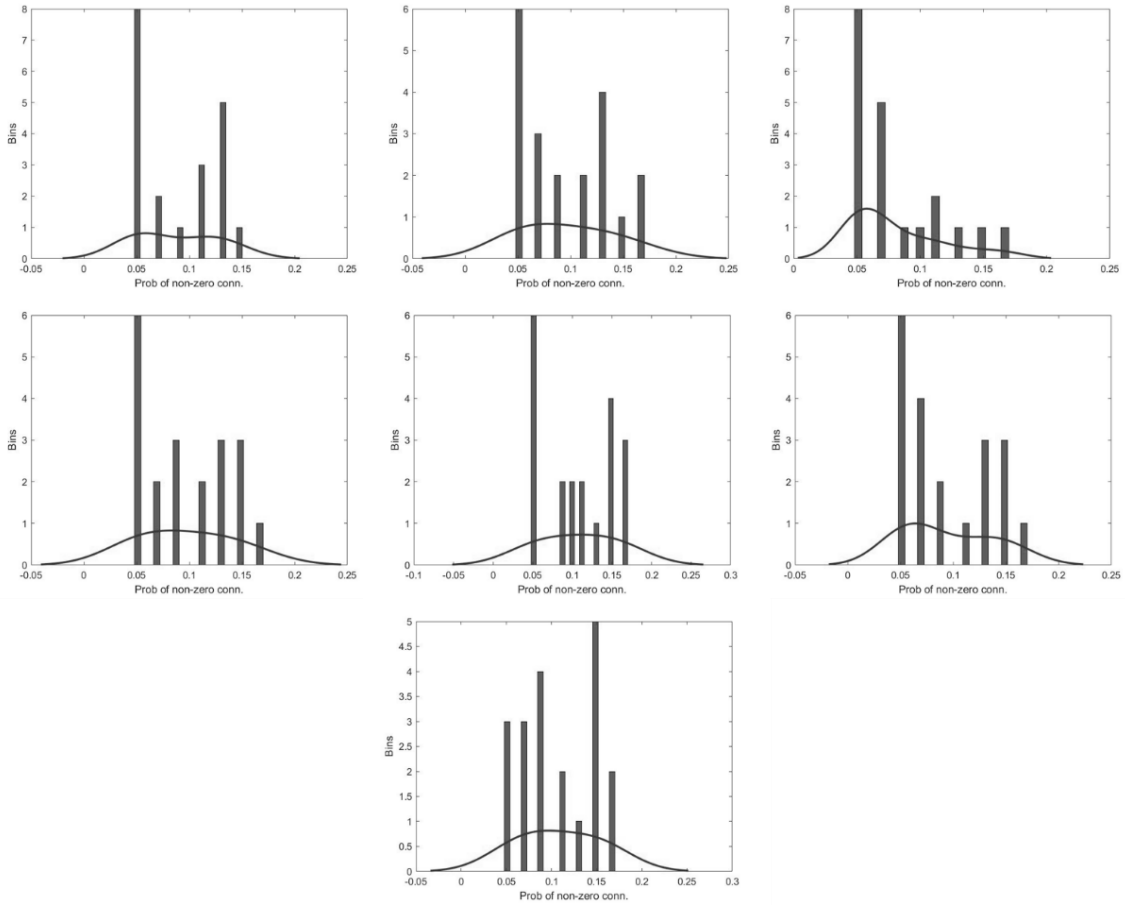

Figure 4.13: Histogram and probability distribution plot for Probability of Non-Zero Connections (Hyperparameter) for Cases A through G for all Attackers and Defenders combined. Cases A through G are arranged in a row-wise sequence

Number (Fig. 4.11), Leak Rate (Fig. 4.12), and Probability of non-zero connections (Fig. 4.13),

respectively. The descriptive statistics of these hyperparameters can be found in Table 4. The probability density plots of these histograms revealed that the distributions are slightly multimodal, further strengthening those mere approximations that cannot be considered while assigning the hyperparameters to perform the prediction task.

4.5. Conclusion

In the current work, a novel massively parallel hybrid ESN was developed to predict the dynamics of every agent in an interacting Adversarial Swarm. Every agent in the system was assigned to a hybrid ESN which was trained and tested separately. The output of the ESNs at every time step during the training and the prediction phase as passed through an external module termed as the discriminator, which was pre- and post-processing the spatio-temporal data. The following are the main conclusions of the current work

- The massively parallel hybrid ESN setup was demonstrated to be a well-suited tool for tackling the prediction of multiagent dynamics.
- The massively parallel hybrid ESN setup developed, and its external module are generic enough to easily be modified to predict a wide range of engineering problems.
- The massively parallel ESN validated the concept of the Individuality of agents in the multiagent system as every agent was mapped uniquely by the ESN.
- The concept of Individuality will pave the way for further studies in this field, which will enable effective control and use of multiagent systems.

REFERENCES

1. Selvi, V., and Umarani, R. "Comparative analysis of ant colony and particle swarm optimization techniques," *International Journal of Computer Applications* Vol. 5, No. 4, 2010, pp. 1-6.

2. Carere, C., Montanino, S., Moreschini, F., Zoratto, F., Chiarotti, F., Santucci, D., and Alleva, E. "Aerial flocking patterns of wintering starlings, Sturnus vulgaris, under different predation risk," *Animal Behaviour* Vol. 77, No. 1, 2009, pp. 101-107.

3. Calovi, D. S., Lopez, U., Ngo, S., Sire, C., Chaté, H., and Theraulaz, G. "Swarming, schooling, milling: phase diagram of a data-driven fish school model," *New journal of Physics* Vol. 16, No. 1, 2014, p. 015026.

4. Perea, L., Gómez, G., and Elosegui, P. "Extension of the Cucker-Smale control law to space flight formations," *Journal of guidance, control, and dynamics* Vol. 32, No. 2, 2009, pp. 527-537.

5. Edwards, S. J. "Swarming on the Battlefield: past, Present, and Future." RAND NATIONAL DEFENSE RESEARCH INST SANTA MONICA CA, 2000.

6. al-Rifaie, M. M., Aber, A., and Raisys, R. "Swarming robots and possible medical applications," 2011.

7. Bailey, I., Myatt, J. P., and Wilson, A. M. "Group hunting within the Carnivora: physiological, cognitive and environmental influences on strategy and cooperation," *Behavioral Ecology and Sociobiology* Vol. 67, No. 1, 2013, pp. 1-17.

8. Stanford, C. B. "The influence of chimpanzee predation on group size and anti-predator behaviour in red colobus monkeys," *Animal Behaviour* Vol. 49, No. 3, 1995, pp. 577-587.

9. Scheel, D. "Profitability, encounter rates, and prey choice of African lions," *Behavioral Ecology* Vol. 4, No. 1, 1993, pp. 90-97.

10. Vaughn, R. L., Shelton, D. E., Timm, L. L., Watson, L. A., and Würsig, B. "Dusky dolphin (Lagenorhynchus obscurus) feeding tactics and multi-species associations," *New Zealand Journal of Marine and Freshwater Research* Vol. 41, No. 4, 2007, pp. 391-400.

11. Nøttestad, L., Fernö, A., and Axelsen, B. E. "Digging in the deep: killer whales' advanced hunting tactic," *Polar Biology* Vol. 25, No. 12, 2002, pp. 939-941.

12. Soham Gupta, J. B., Pradeep Sapkota. "Use of Hybrid Echo State Networks on the Prediction of Group Dynamics of an Adversarial Swarm System," *Preprint*, 2021.

13. Gupta, S., and Baker, J. "Adversarial Swarms as Dynamical Systems," *arXiv preprint arXiv:2105.13294*, 2021.

14. Lu, Z., Hunt, B. R., and Ott, E. "Attractor reconstruction by machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 6, 2018, p. 061104.

15. Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., and Koumoutsakos, P. "Forecasting of spatio-temporal chaotic dynamics with recurrent neural networks: A comparative study of reservoir computing and backpropagation algorithms," *arXiv preprint arXiv:1910.05266*, 2019.

16. Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Physical review letters* Vol. 120, No. 2, 2018, p. 024102.

17. Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B. R., Girvan, M., and Ott, E. "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 4, 2018, p. 041101.

18. Lu, Z., Pathak, J., Hunt, B., Girvan, M., Brockett, R., and Ott, E. "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 27, No. 4, 2017, p. 041102.

19. Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., and Ott, E. "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 27, No. 12, 2017, p. 121102.

20. Chattopadhyay, A., Hassanzadeh, P., Palem, K., and Subramanian, D. "Data-driven prediction of a multi-scale Lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ANN, and RNN-LSTM," *arXiv preprint arXiv:1906.08829*, 2019.

21. Zimmermann, R. S., and Parlitz, U. "Observing spatio-temporal dynamics of excitable media using reservoir computing," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 4, 2018, p. 043118.

22. Canaday, D., Griffith, A., and Gauthier, D. J. "Rapid time series prediction with a hardware-based reservoir computer," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 28, No. 12, 2018, p. 123119.

23. Krishnagopal, S., Girvan, M., Ott, E., and Hunt, B. R. "Separation of chaotic signals by reservoir computing," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 2, 2020, p. 023123.

24. Carroll, T. L. "Mutual Information and the Edge of Chaos in Reservoir Computers," *arXiv preprint arXiv:1906.03186*, 2019.

25. Mitchell, M., Hraber, P., and Crutchfield, J. P. "Revisiting the edge of chaos: Evolving cellular automata to perform computations," *arXiv preprint adap-org/9303003*, 1993.

26. Crutchfield, J. P. "Between order and chaos," *Nature Physics* Vol. 8, No. 1, 2012, pp. 17-24

27. Carroll, T. L. "Dimension of reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 1, 2020, p. 013102.

28. Carroll, T. L., and Pecora, L. M. "Network structure effects in reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 29, No. 8, 2019, p. 083130.

29. Zhang, C., Jiang, J., Qu, S.-X., and Lai, Y.-C. "Predicting phase and sensing phase coherence in chaotic systems with machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 30, No. 8, 2020, p. 083114.

30. Chang, S., and Lathrop, D. "Using reservoir computing to predict temperature fluctuations in turbulent Rayleigh-Bénard convection," *APS Division of Fluid Dynamics Meeting Abstracts*. 2019, p. NP05. 008.

31. Bergstra, J., and Bengio, Y. "Random search for hyper-parameter optimization," *Journal of machine learning research* Vol. 13, No. 2, 2012.

32. Gould, H., and Tobochnik, J. *An Introduction to Computer Simulation Methods: Applications to Physical Systems*: Addison-Wesley, 1996.

33. Ahmed, M. U., and Mandic, D. P. "Multivariate multiscale entropy: A tool for complexity analysis of multichannel data," *Physical Review E* Vol. 84, No. 6, 2011, p. 061918.

34. Costa, M., Goldberger, A. L., and Peng, C.-K. "Multiscale entropy analysis of biological signals," *Physical review E* Vol. 71, No. 2, 2005, p. 021906.

35. Lukoševičius, M. "A practical guide to applying echo state networks," *Neural networks: Tricks of the trade*. Springer, 2012, pp. 659-686.

36. Lukoševičius, M., and Jaeger, H. "Reservoir computing approaches to recurrent neural network training," *Computer Science Review* Vol. 3, No. 3, 2009, pp. 127-149.

37. Shofer, N. "Reservoir Computing: Memory, Nonlinearity, and Spatial Observers." Reed College, 2018.

38. Jaeger, H., and Haas, H. "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *science* Vol. 304, No. 5667, 2004, pp. 78-80.

39. Dalcín, L., Paz, R., Storti, M., and D'Elía, J. "MPI for Python: Performance improvements and MPI-2 extensions," *Journal of Parallel and Distributed Computing* Vol. 68, No. 5, 2008, pp. 655-662.

CHAPTER 5

CONCLUSION

The current work investigated the dynamics of complex adaptive systems in the form of an interacting Adversarial swarm system and studied the predictability of such systems using Machine Learning. The work also explored the idea of individuality in a multiagent system, where each individual agent reacts slightly differently when a uniform stimulus is applied. The work was divided into three main phases to achieve the goals and objectives of this investigation. The following are the conclusions of the first investigation:

- A generic semi-hybrid dynamical system-based Agent-based Adversarial Swarm model was developed using a rigorous numerical procedure. This model is generic enough to adapt to various engineering applications with minor changes to force terms or potential functions.
- In the spectrum of the study conducted for the various population of agents making up each swarm, some test cases revealed transient chaotic behavior with multiple local equilibrium points for both swarms that reaffirmed the existence of complex dynamics.
- The LLEs found out for the Attackers and Defenders for all the trials, respectively, revealed the existence of both chaotic and non-chaotic solutions. The average LLEs indicated that swarms were both on the 'Edge of chaos,' further strengthening the presence of complex dynamical behavior. The Recurrence plots also backed up the presence of chaotic and non-chaotic behavior.
- Finally, Multiscale Entropy (MSE) was evaluated for the center of mass time-series for the swarm from scales 1 to 20. MSE, for both Attackers and Defenders, revealed the existence

of complex dynamics. The MSE revealed an intermediate level of randomness for the simulation overall, thus proving the effectiveness of the Semi-Hybrid approach assumed in the Adversarial swarm model.

In the second and third investigations conducted, ESNs were found to be a useful tool that could be effectively used to predict a highly nonlinear dynamical system. The following were the conclusions from the second investigation:

- Among the two configurations developed in the second investigation, Config 2 showed better overall performance when compared to Config 1 to achieve the prediction task.
- Config 2 achieved a better prediction. It consisted of two parallel independent ESNs that successfully learned the dynamics of individual swarms that are essentially independent of each other and are only coupled by external rules of engagement.
- The LLEs obtained from the ESNs and phase space reconstruction were in good agreement with each other, further implying the success of the ESN to map the interacting swarm dynamics successfully.

In the third and final phase of this study, the following were the main conclusions:

- The massively parallel hybrid ESN setup was demonstrated to be a well-suited tool for tackling the prediction of multiagent dynamics.
- The current massively setup was developed, and its external module was generic enough to easily be modified to predict a wide range of engineering problems.
- The massively parallel ESN validated the concept of the Individuality of agents in the multiagent system as every agent was mapped uniquely by the ESN.
- Individuality is a factor that must be considered for designing large-scale time-series prediction frameworks using machine learning.

- Overall, ESNs proved to be a valuable technique to predict short-term dynamics of interacting multiagent dynamical systems in various engineering applications.

Many future works can be conducted based on the current investigations t continue the study in this field. In the present work, the simulations only considered the interaction between the Attacker and Defender swarm. The simulations were performed in an ideal environment where the interactions occur; noise effects are not explicitly modeled but implicitly considered using random initial conditions. The work also assumed ideal instantaneous inter and intra swarm communication and did not consider the impact of communication delays. The position of the Goal is also assumed to be fixed for all the cases studied, and the work does not assume a moving goal. The Attackers and Defenders are also globally coupled without explicit cut-offs, which is sometimes the case for autonomous agents. The Morse force implicitly imposes the limitation of the range of sensing or vision for the agent. Delay in communication between the agents is not considered, which is often the real-world case due to various factors. Some of these limitations will serve as the basis of future works for the first phase. The second and third phases of the work can be further advanced by training the ESN configurations developed on real-world data obtained from various engineering applications in different fields, from healthcare, biology, and finance to security.

The prediction of the dynamics of the Adversarial Swarm in the future may also be attempted by using Convolutional Neural Networks (CNNs) by taking the spatial screenshots of the agents in regular intervals of the simulation and using the image to ultimately solve a multiclass classification problem or a binary classification problem to identify the final states of the system effectively.