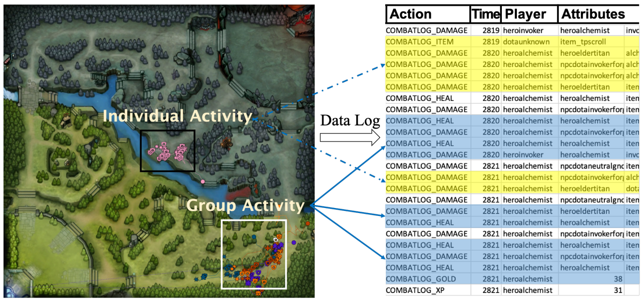
Proceedings of the Seventeenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment
(AIIDE 2021)

Hierarchical Dual Attention-Based Recurrent Neural Networks for Individual and Group Activity Recognition in Games

Sabbir Ahmad,[1] **Magy Seif El-Nasr,** [2] **Ehsan Elhamifar** [1]
[1] Northeastern University, Boston, Massachusetts
[2] University of California Santa Cruz, Santa Cruz, California
ahmad.sab@northeastern.edu, mseifeln@ucsc.edu, eelhami@ccs.neu.edu

Abstract

We study the problem of simultaneously recognizing complex individual and group activities from spatiotemporal data in games. Recognizing complex player activities is particularly important to understand game dynamics and user behavior having a wide range of applications in game development. To do so, we propose a novel framework by developing a hierarchical dual attention RNN-based method that leverages feature and temporal attention mechanisms in a hierarchical setting for effective discovery of activities using interactions among individuals. We argue that certain activities have dependency on certain features as well as on temporal aspects of the data which can be leveraged by our *dual-attention* model for recognition. To the best of our knowledge, this work is the first to address activity recognition using spatiotemporal data in games. In addition, we propose using game data as a rich source of obtaining complex group interactions. In this paper, we present two contributions: (1) two annotated game datasets that consist of individual and group activities, (2) our proposed framework improves the state-of-the-art recognition algorithms for spatiotemporal data by experiments on these datasets.



Action	Time	Player	Attributes	
COMBATLOG_DAMAGE	2819	heroinvoker	heroalchemist	invc
COMBATLOG_ITEM	2819	dotaunknown	item_tpscroll	
COMBATLOG_DAMAGE	2820	heroalchemist	heroeldertitan	alch
COMBATLOG_DAMAGE	2820	heroalchemist	npcdotainvokerfor	alch
COMBATLOG_DAMAGE	2820	heroalchemist	npcdotainvokerfor	alch
COMBATLOG_DAMAGE	2820	heroalchemist	heroeldertitan	iten
COMBATLOG_HEAL	2820	heroalchemist	heroalchemist	iten
COMBATLOG_DAMAGE	2820	heroalchemist	npcdotainvokerfor	iten
COMBATLOG_HEAL	2820	heroalchemist	heroalchemist	iten
COMBATLOG_DAMAGE	2820	heroalchemist	npcdotainvokerfor	iten
COMBATLOG_HEAL	2820	heroalchemist	heroalchemist	iten
COMBATLOG_DAMAGE	2820	heroinvoker	heroalchemist	invc
COMBATLOG_DAMAGE	2821	heroalchemist	npcdotaneutralgnc	iten
COMBATLOG_DAMAGE	2821	heroalchemist	npcdotainvokerfor	alch
COMBATLOG_DAMAGE	2821	heroalchemist	heroeldertitan	dota
COMBATLOG_DAMAGE	2821	heroalchemist	npcdotaneutralgnc	iten
COMBATLOG_DAMAGE	2821	heroalchemist	heroeldertitan	iten
COMBATLOG_HEAL	2821	heroalchemist	heroalchemist	iten
COMBATLOG_DAMAGE	2821	heroalchemist	npcdotainvokerfor	iten
COMBATLOG_HEAL	2821	heroalchemist	heroalchemist	iten
COMBATLOG_DAMAGE	2821	heroalchemist	npcdotainvokerfor	iten
COMBATLOG_HEAL	2821	heroalchemist	heroalchemist	iten
COMBATLOG_GOLD	2821	heroalchemist	38	
COMBATLOG_XP	2821	heroalchemist	31	

(a) Activities on a 2D Map (b) Spatiotemporal Data

Figure 1: Individual and Group activities of multiple actors and spatiotemporal representation of the corresponding activity data.

1 Introduction

Research in games has particularly focused on understanding player strategy and activities in the recent years due to the advancement in game development and game data science. Various methods, including machine learning and statistical analysis, have been proposed for game outcome prediction [Semenov et al. 2016; Ong, Deolalikar, and Peng 2015], understanding individual and team mechanics [Weber and Mateas 2009; Cavadenti et al. 2016; Yang, Harrison, and Roberts 2014; Mahlmann, Schubert, and Drachen 2016] as well as clustering players based on roles and playing patterns [Nascimento Junior et al. 2017; Neidhardt, Huang, and Contractor 2015; Drachen et al. 2014]. However, few works have focused on complex activities and decision makings [Weber and Mateas 2009; Ahmad et al. 2019]. Nevertheless, such proposed methods tend to be game specific or do not scale to large datasets. In this paper, we focus on this under-explored area. Understanding complex activities facilitates identifying behaviors, tactics and strategies allowing game developers and designers adjust games to target audience as well as find imbalance in game mechanics.

Activity recognition is an active area of research in computer vision with over decades of research and applications in security, health, entertainments and sports, among others. Depending on the type of interactions in data, activities can be divided into two groups. The first is the activity performed by an individual person (e.g., tightening a screw, pouring oil) independent of the context. The second is the group activity, which consists of collective activities of multiple humans towards achieving a high-level goal (e.g., assembling a device, cooking a recipe) [Wang, Ni, and Yang 2017; Li and Choo Chuah 2017; Hussein, Gavves, and Smeulders 2019]. While the majority of existing works have focused on understanding individual activities, more recently group activity recognition has gained more attention due to its importance and utility for applications in security, surveillance and entertainments [Ibrahim et al. 2016; Qi et al. 2018; Azar et al. 2019; Gavrilyuk et al. 2020].

We aim to tackle the problem of activity recognition from game data in both individual and group settings simultaneously. An example scenario is illustrated in Figure 1a where a 2D map of a multiplayer game is shown along with player activities for a certain time period. An individual activity is performed by a single player (black rectangle on the map), and a group activity is performed by multiple players (white rectangle). The black rectangle corresponds to "Farm" activity which consists of movement, damage and killing actions by a single player, whereas, the white rectangle corresponds to "Group Fight" which includes actions of multiple players. One major problem of activity recognition from the game video is that the video frame is different for every player and each player can only see a certain region surrounding the current position on the map. Therefore, the entire map for all activities is not present. Instead of the video, we can use log data for recognizing activities, which contains low-level actions of the players (Figure 1b). The log data is spatiotemporal as it contains spatial information of the players for every time-step. Each activity on the map corresponds to certain entries in the log data. For example, the individual and group activity on the map in Figure 1a correspond to "Farm" and "Group Fight", shown by the yellow and blue highlighted entries, respectively.

In this paper, we discuss two contributions. First, we gathered and annotated two datasets for individual and group activities, which we will make available upon request. Second, we propose a novel hierarchical attention-based Recurrent Neural Network (RNN) method for group activity recognition from spatiotemporal game data, using both feature and temporal attentions. For annotating the datasets, we use the work of [Ahmad et al. 2019], which proposed a framework for labeling spatiotemporal game data, applying it to two electronic games: BoomTown and DotA 2. The authors have demonstrated complex individual and group activity labels for the games that can be used to understand player strategies and decision making. We consider the same labels proposed in [Ahmad et al. 2019] for the two activity datasets we use in this paper.

The organization of the paper is as follows. In Section 2, we provide a brief overview of game analytics and activity recognition literature. In Section 3, we demonstrate our proposed models for the group activity recognition. In Section 4, we evaluate the proposed models on two game datasets. Finally, Section 5 concludes the paper.

2 Related Work

The main focus of this work is to recognize individual and group activities using spatiotemporal data in games. Our work is related to game analytics for complex activity recognition. In this section, we give a brief overview of the game analytics and activity recognition tasks.

Game Analytics. Research in games explored various aspects including player strategies, outcome prediction, behavior classification. Plan and goal recognition [Sukthankar et al. 2014] was used to understand player strategies by utilizing simple case-based reasoning to complex plan libraries. Sukthankar et al. [2014] and Min et al. [2016; 2014] explored plan and goal recognition in single player games which focus on recognizing whether the player has attained some fixed goals to complete the game. In addition, several works use machine learning models to predict game outcome [Semenov et al. 2016; Ong, Deolalikar, and Peng 2015; Summerville, Cook, and Steenhuisen 2016] and performance [Miller and Crowcroft 2010; Sapienza et al. 2018; Mahlmann, Schubert, and Drachen 2016]. Other works [Nascimento Junior et al. 2017; Drachen et al. 2012] explored clustering based on the roles and behaviors of the players using clustering algorithms on performance metrics, actions, playing patterns and different attributes. On the other hand, few works [Weber and Mateas 2009; Yang, Harrison, and Roberts 2014] focused on investigating strategic gameplay in individual and group settings. These works are different from the activity recognition task we present here as they lack complex interaction among multiple players. Ahmad et al. [2019] proposed a methodology to visualize complex activities and decision makings in games. In this paper, we use deep models to recognize the complex activities from gamedata.

There has been a great degree of research in games using Reinforcement Learning (RL), multi-agent systems and graphical models. The multi-agent related works [Singh et al. 2017; Demiris 2007] including RL are interesting for simulation and activity detection, however, it is difficult and takes expertise to mathematically formulate the activities and interactions of the complex games we are currently investigating. While the strategic activities can be achieved by RL (as in AlphaStar [Vinyals et al. 2019]), they are hard to formulate mathematically to enable an algorithm to infer or detect these patterns from data and thus infer strategies, which is our goal. Our proposed models take game replay data that are easy to provide without much expertise. Zha et al. [2013] and Lorthioir et al. [2020] proposed graphical model based activity recognition where the use-cases are much simpler than the strategic activities in modern games. The combat models proposed by Uriarte et al. [2015] for two-player RTS games are harder to generalize in multiplayer settings with complex interactions. Moreover, these works are still in research-stage and cannot be applied to complex multiplayer games for analysis. Our proposed models can be applied to sequential data for any type of game setting given that the activity labels are available. The unsupervised method for activity detection proposed by Freedman et al. [2015] is one direction of our future work which can alleviate the burden of labeling.

Activity Recognition. There are several works related to Activity Recognition in Computer Vision. As the input source is image or video, a general approach for activity recognition is to use CNN for extracting features from image or video, and then leverage a recurrent architecture to model the actor dynamics. Wang et al. [2017] used LSTM based networks on top of CNN in a similar manner. Li et al. [2018] proposed Attention based LSTM to focus on particular image locations while Wang and Gupta [2018] used space-time region graphs in long range videos. Girdhar et al. [2019] adapted transformer-style architecture to aggregate features of the actions. These works focused on the individual activity recognition.

Additionally, researchers used sensor data to perform activity recognition using data from wearable devices or smart environment settings. One of the most used methods for these types of temporal data is pattern mining to find the frequent action sequences in the data and then determine the activity from the pattern [Liu et al. 2016]. Besides, deep neural networks were used for human activity recognition from sensor data [Cheng et al. 2018; Rokni, Nourollahi, and Ghasemzadeh 2018]. All of these works used temporal data that are from sensors of an individual user. In our work, we use spatial-temporal data for activity recognition in individual and group settings.

Group Activity Recognition. In recent times, group activity recognition has gained much attention from the researchers in Computer Vision. Deep learning has been heavily used for understanding group dynamics where CNNs are used to extract spatial features from video frames. On top of CNN, recurrent architectures are used to understand the sequential actions of the actors. Ibrahim et al. [2016] proposed a hierarchical LSTM model on top of CNN to determine group activities. Other works explored deep neural network based hierarchical graphical models [Deng et al. 2015], graph convolutional neural network to find actor relationship [Wu et al. 2019] and pose network [Gavrilyuk et al. 2020; Vahora and Chauhan 2019]. Qi et al. [2018] demonstrated a spatiotemporal attention based neural network with semantic graphs to determine group activities.

In our work, we use LSTM based RNN to solve the problem of activity recognition from spatiotemporal data. We also demonstrate variants of the basic LSTM model with attention mechanism and hierarchical architecture to understand group dynamics.

3 Proposed Approach

In this section, we develop a hierarchical attention-based mechanisms based on RNN for the group activity recognition problem from spatiotemporal data. Our goal is to recognize the complex activities performed by the actors individually and as a group. We use spatiotemporal data as input which is a sequence of low-level actions including features of the particular action (see Figure 1b). The output is the probability distribution of a fixed set of activities performed by the players at each time-step of the input sequence.

Formally, at time-step t, we denote the input by $x_t \in \mathbb{R}^d$, with d being the number of features. The input to our RNN-based model, shown in Figure 2a, is a sequence of inputs x_t, and the length of the sequence T is a hyper-parameter. Hence, the input to the model is of shape $T \times d$ for a single actor. The output of the model is of shape $T \times |C|$, where C is the set of activity label classes. $y_t \in \mathbb{R}^{|C|}$ is the output probability distribution over the activity label set at time-step t. Note that C is different for individual and group activity labels. For group activity recognition we also propose a hierarchical model as shown in Figure 2b. This model takes multiple actor data simultaneously, and detects the group dynamics thorugh the hierarchical structure.

A naive approach to address the problem would be to use a Long-Short Term Memory (LSTM) model to extract features and produce outputs,

$$h_t = \mathrm{LSTM}(x_t), \;\; y_t = \mathrm{softmax}(W_{fc} h_t + b_{fc}),$$

where $h_t \in \mathbb{R}^{h_N}$ denotes the hidden state of the LSTM and W_{fc}, b_{fc} denote the weights and biases of a fully connected network. However, such a naive approach ignores interactions between individuals as well as importance of different features and individual actions at different times for group activity recognition. Next, we propose a sequence of steps to build a framework that overcomes these issues.

3.1 Feature Attention LSTM

We have observed that often certain features are more informative for predicting the individual and group activities. For example, "Exploring" (which refers to roaming around the map without taking any action) depends mostly on movements of the players, whereas, "Explosive Inner" (which refers to placing an explosive inside rocks by creating a tunnel) depends on the use and type of explosives. Therefore, to improve the performance, we use a *feature attention* module to a basic deep LSTM model. The feature attention module gives attention to particular features of each step of the input sequence. Replacing the blue highlighted rectangle with a deep LSTM in Figure 2a gives the Feature Attention LSTM. The feature attention module is essentially a fully-connected neural network that takes the sequence data as input and outputs a vector of weights of the features for each time-step of the input sequence. For each step of the sequence, the feature weights sum to 1. This weight is multiplied with the actual input and this multiplied input is given to the deep LSTM. As a result, the LSTM gets the weighted sequence of input from the attention module. At time-step t, the model operations can be formulated as follows.

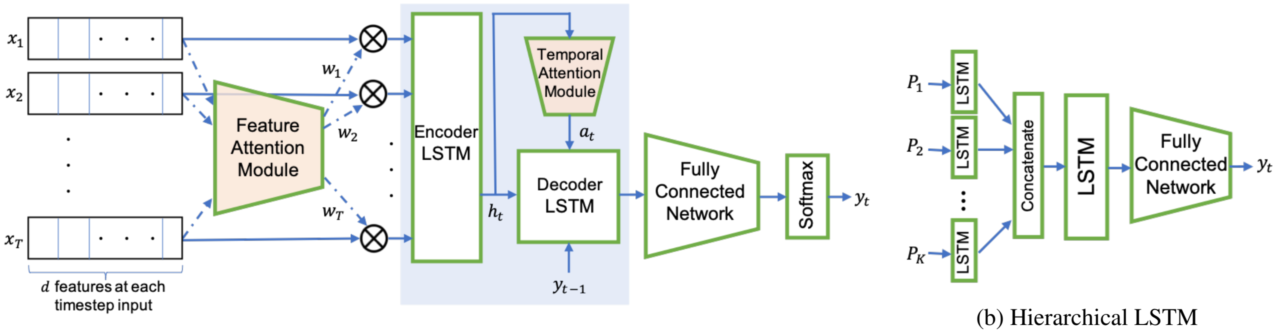
$$w_t^f = \mathrm{feature\_attention}(x_t), \text{where } \textstyle\sum_{i=1}^d w_{ti}^f = 1,$$

$$\tilde{x_t} = w_t^f \odot x_t.$$

Here, the $\mathrm{feature\_attention}$ is a fully connected neural network that outputs weights for the features of the input data, $w_t \in \mathbb{R}^d$ is the output weight. This weighted $\tilde{x_t}$ is used as the input to the deep LSTM.

3.2 Temporal Attention LSTM

We also observe that certain actions at certain time-instants are more informative for recognizing activities. For example, "Explosive Inner" happens when a player creates a tunnel inside rocks and places an explosive. Therefore, destroying rocks for the tunnel as well as the use of explosives are more informative than other actions. Hence, similar to the feature attention LSTM model, we can also give different weights to the temporal step of the hidden representations of the input sequence. In other words, we add a *temporal attention* module to the basic deep LSTM model that gives attention to the steps of the sequence of input. We use ideas from machine translation model by Bahdanau et al. [2014]. The LSTM model here is encoder and decoder based. The encoder gets the input from data and generates hidden representations. The attention module gets the input from the hidden representations and generates a weight vector. The decoder of the LSTM model takes the weighted attention vector and the hidden representations as input. At time-step



(a) Feature and Temporal Attention LSTM Model

(b) Hierarchical LSTM

Figure 2: Overview of the proposed Attention based LSTM models. (a) This diagram shows the Feature and Temporal Attention LSTM model, where the Feature and Temporal Attention modules are fully connected layers. Removing the Attention modules from the diagram converts it into basic Deep LSTM model. Removing the Temporal Attention module gives the Feature Attention LSTM model. In both cases, the encoder-decoder LSTM can also be replaced by a single Deep LSTM. (b) This diagram shows the basics of Hierarchical LSTM models. The first stage LSTMs detects the person level dynamics and the second stange LSTM detects the group dynamics.

t, the model operations can be formulated as follows.

$$w_t = \text{temporal\_attention}(h_t), a_t = w_t \odot h_t$$

$$y_t = \text{DecoderLSTM}(h_t, a_t, y_{t-1})$$

Here, h_t is the hidden representation from the Encoder LSTM, $w_t \in \mathbb{R}^{h_N}$ is the output weight from the temporal attention module which is multiplied with the hidden representation to get the attention weights $a_t \in \mathbb{R}^{h_N}$. This attention weights with hidden representation from Encoder LSTM along with the previous time-step output y_{t-1} is given as input to the Decoder LSTM. The previous time-step output y_{t-1} is used as a teacher forcing fashion during the training with a 50% probability of choosing between the actual output and the model output.

3.3 Feature and Temporal Attention LSTM

We merge the Feature Attention LSTM and the Temporal Attention LSTM to get a *dual-attention* model as shown in Figure 2a. In this model, there are two different attention modules - (i) Feature Attention module gives different weights to every step of the input sequence, and (ii) Temporal Attention module gives different weights to the hidden representations of the input sequences.

3.4 Hierarchical LSTM

We use a hierarchical LSTM model, which allows us to jointly reason about the actions of multiple players that define the group dynamics. The higher level LSTM handles the hidden representations of the individual players simultaneously which is necessary to understand the group interactions as shown in [Ibrahim et al. 2016]. In our case, we have two levels of LSTM models as shown in Figure 2b. The first level identifies the person level interaction and the second level aggregates the person interactions to determine the group activities. The main difference between these two models are that in [Ibrahim et al. 2016] each frame is labeled with a single group activity, whereas in our case, the actors present in the data may have different group activities in a single time-step. The input to the group level LSTM at time-step t can be formulated as follows.

$$y_t^k = h_t^k \oplus x_t^k$$

$$y_t^h = y_t^1 \oplus y_t^2 \oplus \cdots \oplus y_t^K$$

Here, y_t^k is the concatenated sequence of hidden representations and the input features for person k, and y_t^h is the concatenated input for K persons at time-step t. This input is used in a similar fashion for the group level LSTM.

3.5 Hierarchical Feature Attention LSTM

We add the feature attention module with the hierarchical LSTM model. The feature attention is given to the person level interaction detection as in the Feature Attention LSTM model. In Figure 2b, the first stage LSTMs are attached to the feature attention module. The hierarchical part of the model remains the same as the Hierarchical LSTM model which takes input from the first stage, and outputs the probability distribution of the group activities for each actor.

For learning the hierarchical models, the single feature attention LSTM is trained with each actor data separately. We use Cross Entropy Loss function with the individual activity labels to train it. The trained first stage LSTM is attached to the second stage LSTM to train the whole model. This single first stage feature attention LSTM is used multiple times to generate the hidden representation for all the actors present in the data (P_1 to P_K in Figure 2b). The hidden representations are concatenated to each actor data to generate the input for the second stage LSTM. To train the whole hierarchical model, we again use Cross Entropy Loss function with the group activity labels.

3.6 Implementation Details

The person level interaction models are trained in an end-to-end fashion. In the basic deep LSTM and Feature Attention LSTM models we use four layers with 128 hidden units. For the temporal attention models, we use two layers of LSTM units as the temporal attention module adds a very large number of parameters to the models. To balance the number of parameters so that the model does not overfit, we decrease the LSTM layers in the Encoder and Decoder LSTMs. We use teacher forcing in the Temporal Attention models with 50% probability.

The hierarchical models are trained in two steps. We concatenate the trained person level models to the group level LSTM to train the hierarchical models. The hierarchical LSTM consists of a two layer LSTM with 128 hidden units each. All the final layer LSTM models are followed by three fully connected layers which outputs the probability distribution of activity labels for each time-step.

We implemented our models in PyTorch and trained in Google-Colab with Tesla V100 GPU and 26 GB of RAM. We used Adam optimizer with a fixed learning rate of 0.1 to train all the models. For hyper-parameter (e.g. hidden-units, learning-rate, window-size) tuning, we tried different settings and chose the best performing one on validation data for BoomTown. We kept similar settings for DotA2 except for a shorter window-size as the data is more complex. We used dropouts for regularization and weighted cross entropy loss as the label distribution in the datasets are skewed.

4 Experiments

In this section, we describe our datasets and then evaluate our models on the datasets.

4.1 Datasets and Preprocessing

We evaluate the performance of the models using two datasets - BoomTown and DotA 2 that we developed. BoomTown is a multiplayer online team-based game where the goal is to maximize the amount of collected gold in a two-dimensional map. DotA 2 is a popular multiplayer online battle arena (MOBA) game. The goal of this game is to destroy the opponent team's base. In both of the games, the players take different actions individually and as a group to perform various tasks and achieve the goal. The low-level actions taken by the players in the games can be accumulated as complex strategic activities as shown in [Ahmad et al. 2019]. The activity labels for BoomTown and DotA 2 are defined by Ahmad et al. [2019] and in this work, we use the same definitions of activities to annotate the datasets for the individual and group activity recognition.

BoomTown Dataset. This dataset contains 19 matches with a total of 68 players. Each team has 3-6 players and each match is player by a single team. Each match data contains temporal information of each player's low-level actions with related attributes. We preprocess the match data to separate data for each actor in a match. The data is then normalized and converted to one-hot encoding for the categorical values. Finally, there are 28 features with 8 different individual activity labels. The features include low-level action name (e.g. change position, place explosive, etc), position coordinates, timestamp, gold acquired, number of rocks, items used, etc. Out of 19 matches, we randomly chose 4 matches for testing, 4 matches for validation and the remaining 11 matches for training. The distribution of the labels in the BoomTown Dataset is given in Table 1.

BoomTown		DotA 2			
Individual Label	Count	Individual Label	Count	Group Label	Count
Nothing	8127	Nothing	150060	Nothing	199650
Exploring	18366	Farm	16080	Gank	3760
Pickaxing for Gold	20791	Kill	750	Group Fight	4810
Tunneling Pickaxe	1815	Roam	83570	Push	11870
Explosive Inner	16227			Roshan	790
Explosive Tunneling	781			Split Push	4460
Explosive Miss	66			Team Fight	25120
Explosive Quick Mine	6041				

Table 1: Label Distributions in the Datasets

Model	Accuracy
Multi Layer Perceptron	23.3
Logistic Regression	57.5
Support Vector-Machine	68.5
Deep LSTM	73.4
Feature Attention LSTM	74.5
Temporal Attention LSTM	75.1
Feature and Temporal Attention LSTM	75.4

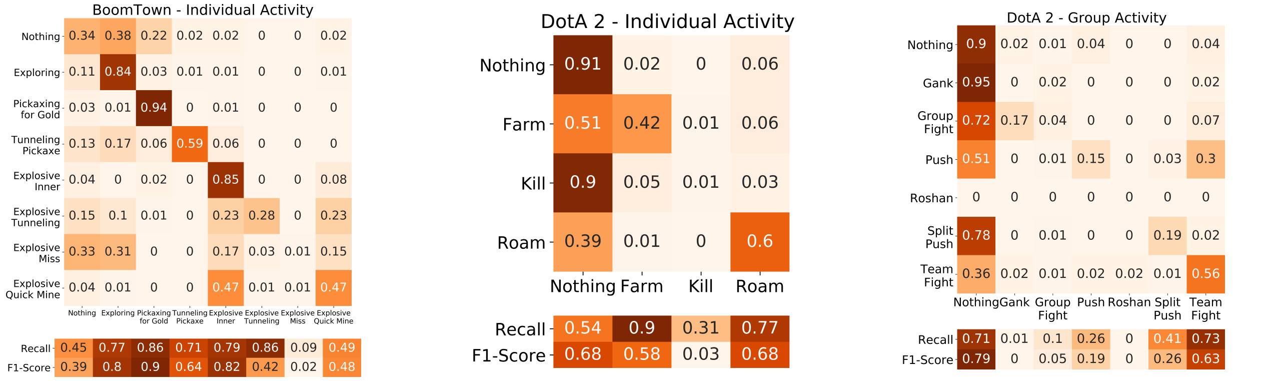
Table 2: Model comparison on the BoomTown Dataset

DotA 2 Dataset. This dataset contains 6 matches with a total of 60 players. Each game is played between two teams of five players each. Similar to BoomTown data, we process each match data to separate individual player data. After normalization and one-hot encoding of the categorical features there are 71 features with 4 individual and 7 group activity labels. The features include low-level action name (e.g. damage, acquire gold, get item, etc), linked unit, health, xp, distance to nearest structure, etc. Out of 6 matches, we randomly chose one match for testing, and the remaining 5 matches for training the models. As the dataset is small, we did not use any validation set. The distribution of the labels in the DotA 2 Dataset is given in Table 1.

4.2 Experiments on BoomTown Dataset

As mentioned previously, the BoomTown Dataset contains only individual activity labels. For this dataset, we use the person level interaction models to compare the performances. We report the accuracy of each model for the individual activity recognition. We also report the accuracy using Multi-Layer Perceptron, Logistic Regression and Support Vector Machine to compare our models to the traditional Machine Learning models.

Notice in Table 2 that our proposed models perform better than the traditional ML models in terms of accuracy. Among the traditional ML models, Support Vector Machine performed closest to our baseline Deep LSTM model with 68.5% accuracy. The Deep LSTM model, which takes the spatiotemporal data directly to the LSTM performs with 73.4% accuracy. The feature attention module gives attention to important features for learning the distribution, which



(a) Feature and Temporal Attention LSTM

(b) Deep LSTM

(c) Hierarchical Feature Attention LSTM

Figure 3: Confusion matrix, recall and f1-score for BoomTown and DotA 2 Datasets using different models

improves the accuracy from the baseline. Using the temporal attention model gives even better accuracy, which can be explained by the fact that the actions taken by the actors have different impacts on the current activity. The temporal attention gives more importance to the low-level actions necessary for the activity. Finally, when the feature and temporal attention modules are aggregated the model performs better than all the models with 75.4% accuracy.

In Figure 3a, the confusion matrix, recall and f1-score for the Feature and Temporal Attention LSTM model are shown. Note that, the main diagonal of the confusion matrix contains the precision values. The model did well on recognizing "Exploring", "Pickaxing for Gold" and "Explosive Inner" having high precision, recall and f1-score. The model got confused between "Nothing" and "Exploring" as both behaviors mostly contain roaming around the map which makes them pretty similar. Likewise, "Explosive Quick Mine", "Explosive Tunneling" and "Explosive Inner" are similar where all these activities contain the use of explosives in a slightly different manners, and the model faced a hard time distinguishing these labels. "Explosive Miss" is a special label which refers to the misuse of an explosive in a random place and appeared in only three matches among the 19 matches. Therefore, the model did not get enough instances to learn the distribution properly. This label also has a short duration (average 7 seconds) compared to others (more than 20 seconds except for "Tunneling Pickaxe").

4.3 Experiments on DotA 2 Dataset

DotA 2 Dataset contains both individual and group activities. We use the person level interaction models as well as the hierarchical models for comparison. We use the person level interaction models to determine both individual and group activities. The hierarchical models are only used for group label recognition. We report the model accuracy for individual and group activity recognition. We also report the accuracy using Logistic Regression. In case of DotA 2, Multi-Layer Perceptron and Support Vector Machine failed to classify all the activity labels except for 'Nothing' label.

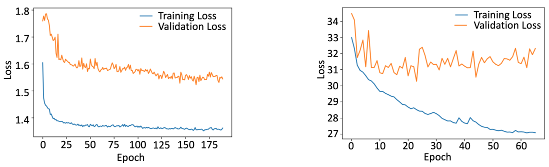
Notice in Table 3 that the Deep LSTM and the Feature Attention LSTM performs similarly on the individual activity recognition, however, Deep LSTM performed much better on group activity recognition. The temporal attention models were unable to learn the activities due to the high skewness of the dataset. We argue that the number of model parameter is very high for the feature and temporal attention models with respect to the size of the dataset, which contributes to the performance drop compared to the baseline. Between the hierarchical models, the Hierarchical Feature Attention LSTM performed better than the Hierarchical LSTM with 66.8% accuracy. Both the hierarchical models performed better than the person-level interaction models in terms of accuracy of the group activity recognition.

Model	Accuracy	
	Individual	Group
Logistic Regression	38.0	41.9
Deep LSTM	63.7	63.2
Feature Attention LSTM	62.9	57.0
Hierarchical LSTM	-	65.3
Hierarchical Feature Attention LSTM	-	66.8

Table 3: Model comparison on the DotA 2 Dataset

The confusion matrix, recall and f1-score of individual activity labels for the Deep LSTM are shown in Figure 3b. Notice that all the activity labels, specially the "Kill" label, are biased to the "Nothing" label. These individual labels, particularly the "Nothing" and "Roam", are very similar in behavior, which makes the model confused. Additionally, the "Kill" label appears comparatively few times compared to other labels. Also the duration of the "Kill" label (average 6 seconds) is significantly very small compared to other labels ("Farm" - 37 seconds and "Roam" - 59 seconds).

The confusion matrix, recall and f1-score of the group activity labels for the Hierarchical Feature Attention LSTM are shown in Figure 3c. One can see a similar biasness to the "Nothing" label due to the very large number of "Nothing" label present in data as shown in Table 1. "Gank" behavior is particularly similar to "Nothing" as both contains roaming around the map with an additional attack scenario in the end



(a) Feature and Temporal Attention LSTM for BoomTown

(b) Hierarchical Feature Attention LSTM for DotA 2

Figure 4: Training and Validation graphs of the models

for "Gank". Additionally, "Gank" and "Group Fight" have comparatively less duration (average 14 and 16 seconds respectively) compared to others (more than 30 seconds). Besides, some labels only appeared in a few matches. For example, "Split Push" (a strategy taken by the players of a team at the same time in different locations on the map), "Roshan" (players engaged a non-player character named Roshan in combat) labels were not present in all the matches.

4.4 Discussion

The data distribution and the confusion matrices show that the models perform better on the labels that have more instances. Though we used a weighted loss function, the models struggled to learn the labels that have very few instances (e.g., "Explosive Miss", "Kill"). Another observation is that the labels having less duration are not well recognized by the models (e.g., "Explosive Miss", "Kill", "Gank"). Besides, some activities (e.g., "Explosive Miss", "Roshan") do not occur in every match, therefore, the corresponding labels are specific to special match conditions and hard to generalize with small number of instances.

Notice that the models perform better on BoomTown than on DotA 2 Dataset. A reason behind that is the fact that DotA 2 is a more complex game where the individual and group activities are composed of very similar low-level actions. Figure 4 shows the training and validation loss with respect to the number of epochs. For Feature and Temporal Attention LSTM on BoomTown Dataset, the training converges in between 30-40 epochs and the validation loss has a similar trend. On the other hand, for Hierarchical Feature Attention LSTM on DotA 2 Dataset, the model overfits very quickly because of a small dataset having complex activity labels. The other models have similar training patterns on both the datasets. Ahmad et al. [2019] reported an Inter-Rater Reliability (IRR) score of 0.95 for BoomTown and 0.72 for DotA 2 in Cohen's Kappa measure [Bordens and Abbott 2002]. The IRR which is a measurement of agreement between labels was calculated between two human labellers. Though IRR is different from the metrics we used here, it showed similar difficulty in recognizing the labels in DotA 2.

5 Conclusion

We studied the problem of individual and group activity recognition in spatiotemporal data settings. We proposed attention-based hierarchical LSTM models for individual and group activity recognition in complex scenarios. We also introduced group activity recognition task in spatiotemporal data which is an unexplored area. We developed two datasets from game data that can be further used to explore activity recognition in games using spatiotemporal data. We plan to make the data available for research based on a contract agreement between different parties that own the data. In future, we aim to tackle the issue of small dataset so that models can learn the dynamics using little amount of data.

Acknowledgements

We would like to acknowledge Defense Advanced Research Projects Agency (DARPA) for funding the work on DotA 2. We further would like to acknowledge Soar Technology, Inc., and the Army for funding the work on data abstraction that resulted in annotating the dataset under Phase II Small Business Innovative Research Grant, contract W56HZV-17-C-0012. We thank Soar Tech for scrapping the DotA 2 data for us from the public replays. We also want to thank Gallup for providing us with the BoomTown data.

References

Ahmad, S.; Bryant, A.; Kleinman, E.; Teng, Z.; Nguyen, T.-H. D.; and Seif El-Nasr, M. 2019. Modeling Individual and Team Behavior through Spatio-temporal Analysis. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, 601–612.

Azar, S. M.; Atigh, M. G.; Nickabadi, A.; and Alahi, A. 2019. Convolutional relational machine for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7892–7901.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Bordens, K. S.; and Abbott, B. B. 2002. *Research design and methods: A process approach*. McGraw-Hill.

Cavadenti, O.; Codocedo, V.; Boulicaut, J.-F.; and Kaytoue, M. 2016. What did i do wrong in my MOBA game? Mining patterns discriminating deviant behaviours. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, 662–671. IEEE.

Cheng, W.; Erfani, S. M.; Zhang, R.; and Kotagiri, R. 2018. Predicting Complex Activities from Ongoing Multivariate Time Series. In *IJCAI*, 3322–3328.

Demiris, Y. 2007. Prediction of intent in robotics and multi-agent systems. *Cognitive processing* 8(3): 151–158.

Deng, Z.; Zhai, M.; Chen, L.; Liu, Y.; Muralidharan, S.; Roshtkhari, M. J.; and Mori, G. 2015. Deep structured models for group activity recognition. *arXiv preprint arXiv:1506.04191* .

Drachen, A.; Sifa, R.; Bauckhage, C.; and Thurau, C. 2012. Guns, swords and data: Clustering of player behavior in computer games in the wild. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 163–170. IEEE.

Drachen, A.; Yancey, M.; Maguire, J.; Chu, D.; Wang, I. Y.; Mahlmann, T.; Schubert, M.; and Klabajan, D. 2014. Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (dota 2). In *2014 IEEE Games Media Entertainment*, 1–8. IEEE.

Freedman, R. G.; Jung, H.-T.; and Zilberstein, S. 2015. Temporal and object relations in unsupervised plan and activity recognition. In *2015 AAAI Fall Symposium Series*.

Gavrilyuk, K.; Sanford, R.; Javan, M.; and Snoek, C. G. 2020. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 839–848.

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A. 2019. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 244–253.

Hussein, N.; Gavves, E.; and Smeulders, A. W. 2019. Timeception for complex action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 254–263.

Ibrahim, M. S.; Muralidharan, S.; Deng, Z.; Vahdat, A.; and Mori, G. 2016. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1971–1980.

Li, X.; and Choo Chuah, M. 2017. Sbgar: Semantics based group activity recognition. In *Proceedings of the IEEE international conference on computer vision*, 2876–2885.

Li, Z.; Gavrilyuk, K.; Gavves, E.; Jain, M.; and Snoek, C. G. 2018. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding* 166: 41–50.

Liu, Y.; Nie, L.; Han, L.; Zhang, L.; and Rosenblum, D. S. 2016. Action2Activity: recognizing complex activities from sensor data. *arXiv preprint arXiv:1611.01872* .

Lorthioir, G.; and Inoue, K. 2020. Design Adaptive AI for RTS Game by Learning Player's Build Order. In *IJCAI*, 5194–5195.

Mahlmann, T.; Schubert, M.; and Drachen, A. 2016. Esports Analytics Through Encounter Detection. Mit sloan sports analytics conference.

Miller, J. L.; and Crowcroft, J. 2010. Group movement in world of warcraft battlegrounds. *International Journal of Advanced Media and Communication* 4(4): 387–404.

Min, W.; Ha, E.; Rowe, J.; Mott, B.; and Lester, J. 2014. Deep learning-based goal recognition in open-ended digital games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 10.

Min, W.; Mott, B. W.; Rowe, J. P.; Liu, B.; and Lester, J. C. 2016. Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks. In *IJCAI*, 2590–2596.

Nascimento Junior, F. F. d.; Melo, A. S. d. C.; da Costa, I. B.; and Marinho, L. B. 2017. Profiling Successful Team Behaviors in League of Legends. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, 261–268. ACM.

Neidhardt, J.; Huang, Y.; and Contractor, N. 2015. Team vs. team: Success factors in a multiplayer online battle arena game. In *Academy of Management Proceedings*, volume 1, 18725. Academy of Management Briarcliff Manor, NY 10510.

Ong, H. Y.; Deolalikar, S.; and Peng, M. 2015. Player Behavior and Optimal Team Composition for Online Multiplayer Games. *arXiv preprint arXiv:1503.02230* .

Qi, M.; Qin, J.; Li, A.; Wang, Y.; Luo, J.; and Van Gool, L. 2018. stagnet: An attentive semantic RNN for group activity recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 101–117.

Rokni, S. A.; Nourollahi, M.; and Ghasemzadeh, H. 2018. Personalized human activity recognition using convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Sapienza, A.; Zeng, Y.; Bessi, A.; Lerman, K.; and Ferrara, E. 2018. Individual performance in team-based online games. *Royal Society Open Science* 5(6): 180329.

Semenov, A.; Romov, P.; Korolev, S.; Yashkov, D.; and Neklyudov, K. 2016. Performance of machine learning algorithms in predicting game outcome from drafts in dota 2. In *International Conference on Analysis of Images, Social Networks and Texts*, 26–37. Springer.

Singh, S.; Lu, S.; Kokar, M. M.; Kogut, P. A.; and Martin, L. 2017. Detection and classification of emergent behaviors using multi-agent simulation framework (WIP). In *Proceedings of the Symposium on Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems*, 1–8.

Sukthankar, G.; Geib, C.; Bui, H. H.; Pynadath, D.; and Goldman, R. P. 2014. *Plan, activity, and intent recognition: Theory and practice*. Newnes.

Summerville, A.; Cook, M.; and Steenhuisen, B. 2016. Draft-Analysis of the Ancients: Predicting Draft Picks in DotA 2 using Machine Learning. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Uriarte, A.; and Ontanón, S. 2015. Automatic learning of combat models for RTS games. In *Eleventh artificial intelligence and interactive digital entertainment conference*.

Vahora, S.; and Chauhan, N. 2019. Deep neural network model for group activity recognition using contextual relationship. *Engineering Science and Technology, an International Journal* 22(1): 47–54.

Vinyals, O.; Babuschkin, I.; Chung, J.; Mathieu, M.; Jaderberg, M.; Czarnecki, W.; Dudzik, A.; Huang, A.; Georgiev, P.; Powell, R.; Ewalds, T.; Horgan, D.; Kroiss, M.; Danihelka, I.; Agapiou, J.; Oh, J.; Dalibard, V.; Choi, D.; Sifre, L.; Sulsky, Y.; Vezhnevets, S.; Molloy, J.; Cai, T.; Budden, D.; Paine, T.; Gulcehre, C.; Wang, Z.; Pfaff, T.; Pohlen, T.; Yogatama, D.; Cohen, J.; McKinney, K.; Smith, O.; Schaul, T.; Lillicrap, T.; Apps, C.; Kavukcuoglu, K.; Hassabis, D.; and Silver, D. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.

Wang, M.; Ni, B.; and Yang, X. 2017. Recurrent modeling of interaction context for collective activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3048–3056.

Wang, X.; and Gupta, A. 2018. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, 399–417.

Weber, B. G.; and Mateas, M. 2009. A data mining approach to strategy prediction. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, 140–147. IEEE.

Wu, J.; Wang, L.; Wang, L.; Guo, J.; and Wu, G. 2019. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9964–9974.

Yang, P.; Harrison, B. E.; and Roberts, D. L. 2014. Identifying patterns in combat that are predictive of success in MOBA games. In *FDG*.

Zha, Y.-B.; Yue, S.-G.; Yin, Q.-J.; and Liu, X.-C. 2013. Activity recognition using logical hidden semi-Markov models. In *2013 10th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 77–84. IEEE.