You're Not Alone in Battle: Combat Threat Analysis Using Attention Networks and a New Open Benchmark (Appendix)

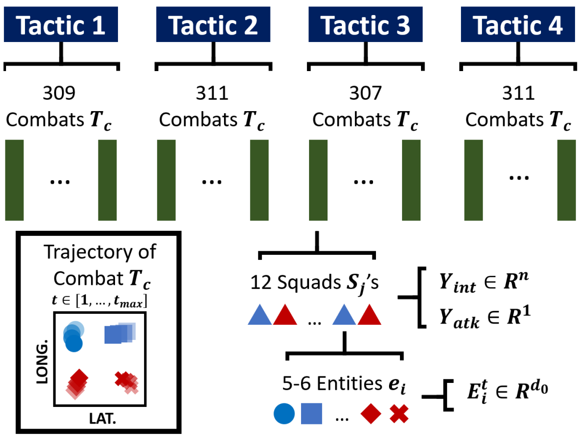

Figure 1: An overview of the proposed data structure.

A DATASET DETAILS

Dataset Structure. We visualize the proposed dataset structure in Figure 1. Combats T_c's are assigned one of four hostile tactics. In each combat T_c, there are 12 squads S_j's, each of which has 5-6 entities e_i's. Labels Y_{int} and Y_{atk} are defined at the squad or squad pair level, respectively, and features E_i^t are defined at the entity level. Over time t, entity features E_i^t change.

Label Distribution. We provide label statistics within each tactic. Table 1 shows the intentions class distribution varies across different tactics, which partially explains the generally low performance of intention prediction (Figure 3-4).

Time Distribution. We provide time distribution within each tactic in Table 2. The total run time does not differ significantly across combats or tactics. The first death in combat on average occurs much earlier than the mean run time.

Feature Distribution. We provide the overall feature distribution. We conflate the time dimension and report its statistics. Table 3 shows that feature scales vary significantly across each dimension. This requires proper scaling for stable model training. The feature distributions across time and tactics are reported in Figure 1 of the main text.

Semantics. In the proposed dataset, all combats T_c's occur in the same geographic region. Each combat T_c has an attacking hostile and a defending friendly force. Not all squads S_j's within each combat necessarily engage in attack, but each of them is assigned an intention Y_{int}. Each entity e_i corresponds to a ground-force soldier. Entities e_i's can die from attacks, resulting in all entity features E_i^t with varying lengths of t.

Also, we provide the detailed meaning of each dimension of E_i^t. The first three dimensions are latitude(degree), longitude(degree), and altitude(meter), respectively. The 4^{th} dimension is attitude: yaw(deg). Speed (km) is the 5^{th} dimension, and the 6^{th} dimension is the force identifier, where 0s correspond to the friendly and 1s to the hostile. The last five dimensions are terrain identifiers for each entity, including Road, Forest, Open Lane, Hiding Place, and Building.

Table 1: Label statistics per tactic

Tactic	Intention Labels #(TE)	#(MT)	#(CR)	#(SS)	#(FE)	#(SP)	Attack Labels #(Attacks)
Linear Advancement (1)	1236	1236	618	0	0	618	1916.
Sequential Progression (2)	4	1244	1862	0	0	622	2110
Flanking Maneuver (3)	0	0	1842	1228	0	614	2455
Direct Engagement (4)	0	0	1866	4	1550	312	3130
All Tactics	1240	2480	6188	1232	1550	2166	9611

• **Abbreviations**: TE = Tactical Engagement. MT = Maneuvering Techniques. CR = Coordinated Rendezvous. SS = Strategic Surprise. FE = Forceful Engagement. SP = Strategic Positioning.

Table 2: Time statistics per tactic (in seconds)

	Mean Run Time	StDev Run Time	Min Run Time	Max Run Time	Mean First Death
Linear Advancement (1)	1227	147	841	2018	564
Sequential Progression (2)	1320	177	1023	2565	571
Flanking Maneuver (3)	1600	179	1209	2345	1042
Direct Engagement (4)	1419	222	964	2191	892
All Tactics	1391	229	841	2565	767

Table 3: Trajectory feature statistics

	Latitude (Degree)	Longitude (Degree)	Altitude (Meter)	Attitude (Yaw)	Speed (Km/h)	Terrain RD	Terrain FR	Terrain OL	Terrain HP	Terrain BD
All Tactics	37.9 ± 0.008	128.1 ± 0.012	667.0 ± 83.1	-27.8 ± 114.4	1.7 ± 2.4	0.221	0.666	0.068	0.040	0.005

• **Abbreviations**: RD = Road. FR = Forest. OL = Open Lane. HP = Hiding Place. BD = Building. **Continuous variables**: Mean ± standard deviation. **Binary variables**: The ratio of positive entries.

B DATASET PREPROCESSING DETAILS

Time Sampling. In sampling timestamps t, we sample from the timestamps less than or equal to $t = 300$. That is, we randomly sample 20 timestamps from $t \in [1, ..., 300]$. We chose a small value of maximum t (300) to predict hostile attacks and intentions much before the first attack occurs.

Standardization. We standardize the input features E_i^t since their scales are highly different across timestamps. For continuous features, we use the overall feature dimension mean μ and standard deviation σ^2 values from the *train set*. Specifically, a normalized continuous feature element is $\frac{x-\mu}{\sigma^2}$, where x is the continuous feature element. Importantly, we do not normalize each combat T_c at an independent scale, since it may cause significant information loss. For example, if we standardize each combat T_c independently, the coordinate feature distribution can be similar between a combat with close entities and one with distant entities. We do not standardize the binary features.

C EXPERIMENTAL DETAILS

Data Split. Table 1 shows that intention class 5 (Forceful Engagement) is only present in Tactic 4 (*Direct Engagement*). Thus, we do not choose Tactic 4 as the test set (recall that test set is composed of only one tactic).

Optimization. All neural network models are optimized with Adam optmizer [2]. All model parameters are trained using a supervised approach, utilizing the cross-entropy loss as the training objective. We use mini-batch training, with batch size of 128. The number of epochs is 50, and the best model is chosen based on the train loss. A small number of epochs is chosen due to the absence of a validation set. We find a larger number of epochs generally do not contribute to performance gain, often leading to overfitting to the train set. For kNN and XG-Boost, we use the default optimization algorithms provided in scikit-learn [3] and XG-boost [1] libraries.

Loss Reweighting. Due to class imbalance in both intention and attack prediction, we reweight the loss based on the train label distribution. Specifically, we choose the reciprocal of the label counts in the train set to reweight the loss.

Hyperparameters. We optimize the hyperparameters with a grid search for neural network models. Specifically, for each model, the dropout rate $\in \{0.3, 0.5, 0.7\}$, learning rate decay $\in \{5e{-}4, 1e{-}4, 5e{-}5, 1e-5, 5e-6, 1e-6\}$, the number of layers $\in \{2, 4\}$ are searched. However, the number of layers for SAFETY was fixed to 2 (1 spatial attention and 1 temporal attention layer). The hidden dimension of 64 and the learning rate of 0.01 are fixed for all neural network models. For kNN and XG-Boost, we use default hyperparameter settings provided in [1, 3].

Implementation Details. For neural network models, Layer Normalization is added to all neural network models, and all models share the same squad aggregation function. However, for the static models, we concatenate the input features' time dimension as the input.

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[2] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[3] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.