GADformer: A Transparent Transformer Model for Group Anomaly Detection on Trajectories

Andreas Lohrer, Darpan Malik, Claudius Zelenka, Peer Kröger
Information Systems and Data Mining, *Kiel University*
Kiel, Germany
{alo,cze,pkr}@informatik.uni-kiel.de, stu225397@mail.uni-kiel.de

***Abstract*—Group Anomaly Detection (GAD) identifies unusual pattern in groups where individual members might not be anomalous. This task is of major importance across multiple disciplines, in which also sequences like trajectories can be considered as a group. As groups become more diverse in heterogeneity and size, detecting group anomalies becomes challenging, especially without supervision. Though Recurrent Neural Networks are well established deep sequence models, their performance can decrease with increasing sequence lengths. Hence, this paper introduces GADformer, a BERT-based model for attention-driven GAD on trajectories in unsupervised and semi-supervised settings. We demonstrate how group anomalies can be detected by attention-based GAD. We also introduce the Block-Attention-anomaly-Score (BAS) to enhance model transparency by scoring attention patterns. In addition to that, synthetic trajectory generation allows various ablation studies. In extensive experiments we investigate our approach versus related works in their robustness for trajectory noise and novelties on synthetic data and three real world datasets.**

***Index Terms*—Group Anomaly Detection, BERT, Model Inspection, Trajectories, Deep Learning, Artificial Intelligence**

I. Introduction

Group Anomaly Detection (GAD) is an important task across many disciplines and domains like computational fluid dynamics and computer vision [1, 2], mobility [3, 4], physics [5, 6], social networks [7] and many more. In these domains, GAD is suitable for various types of group anomalies. Since group member instances can be an arbitrary representation, the GAD paradigm also applies to the detection of anomalous sequences like trajectories, to which [8] refers to as collective anomalies. Especially in the spatio-temporal domain, Trajectory Anomaly Detection is a common task to reveal abnormal behavior as the authors [9, 10, 3] confirm.

However, although sequential coordinates of a trajectory obviously represent a group structure, the detection of individual anomalous trajectories has not been addressed as a group anomaly detection problem yet and not at all with transparent multi-head attention.

The current state of the art approaches for anomaly detection on trajectories are recurrent neural networks (RNNs) like LSTMs [11] and GRUs [12, 10], but the potential of deep learning methods for Group Anomaly Detection has rather been sparsely investigated. So far GAD tasks have more likely been solved by generative topic models [13, 7, 1] or SVM-based methods [14, 5, 4]. Despite recent advances, deep generative models got only involved in form of Adversarial Autoencoders (AAEs) and Variational Autoencoders (VAEs) [2] to perform GAD for images. [3] offers different machine learning based algorithms to detect anomalous groups of multiple trajectories, but does not identify the detection of individual anomalous trajectories (a sequence of group members) as group anomaly detection problem.

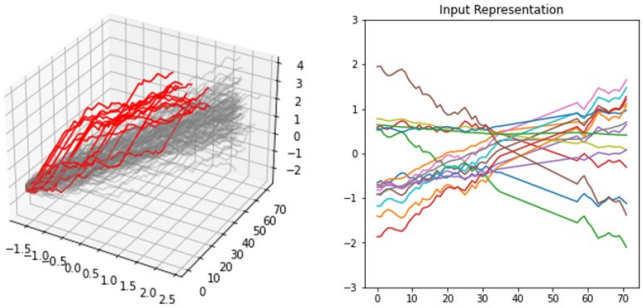

Fig. 1. GADformer trajectory representations on synthetic trajectory data - on the left: 72 raw trajectory steps p_n (=group members o_n) as part of gray normal or red abnormal groups (=individual trajectories Λ_m); on the right: trajectory step embeddings e_n of one individual trajectory (=group $\mathcal{G}$).

However, anomalous behavior is not ensured to just appear within short trajectory segments. It is challenging for recurrent neural networks, sometimes even LSTM and GRU, to learn very long-term dependencies [10].

A further challenge for deep learning based group anomaly detection on trajectories is, that although trajectory data is highly available, it is rather weakly labeled or does not overcome the nonground-truth problem [9] at all.

In order to tackle these challenges we introduce our approach GADFormer, a BERT[15] based architecture with transformer[16] encoder blocks for attention-based group anomaly detection on trajectories. Extending the idea of [16] to optimize also image, audio or video sequence tasks by their transformer approach, we identify transformer based models for a sequence of trajectory points/segments as group member instances of a group anomaly detection task as similarly beneficial. Our model can be trained in an unsupervised as well as in a semi-supervised setting so that there is no or only reduced need for labeled trajectories. Furthermore, we introduce a Block Attention-anomaly Score (BAS), which allows us to provide an transparent view to the capability of the transformer

encoder blocks to distinguish normal from abnormal trajectory attention matrices. We show with extensive experiments on synthetic and real world datasets that our approach is on par with the state of the art methods like GRU or MainTulGAD, an adapted version of [17] for GAD.

Hence, the contributions of our work can be summarized as follows:

- Transformer-Encoder-architecture capable to perform attention-based group anomaly detection in an unsupervised and semi-supervised setting.
- Identification of the detection of individual anomalous trajectories as Group Anomaly Detection problem for BERT based transformer models.
- Block Attention-anomaly Score for group anomalies among aggregated attention pattern of multiple attention heads providing transparency for model inspection.
- Extensive ablation and robustness studies addressing noise, novelties and standard deviation.

The remainder of this work is structured as follows. Section II introduces the a formal description of the addressed problem before in Section III the architecture, training and model transparency of GADFormer is proposed. The experiments in Section IV demonstrate relevance and suitability of our approach across multiple domains and Section V distinguishes our approach from related work. A final summary of the paper as well as an outline to future work is given by Section VI.

II. Preliminaries and Problem Definition

This section provides preliminary terminology and definitions used in this work if not referenced otherwise.

A. Preliminaries

Group Anomaly Detection (GAD) aims to identify groups that deviate from the regular group pattern[2].
Group is a set or sequence of at least two group member instances.
Group Member Instance is an arbitrary data entity described by a n-dimensional feature vector as part of a group.
[Group Anomaly or] Collective Anomaly refers to a collection of data points that belong together and, as a group, deviate from the rest of the data.[8]

B. Problem Definition

The definitions for GAD align with [2] for deep generative models, but got partially a different notation to emphasize its suitability for group anomaly detection on individual trajectories. The GAD problem is described as follows:

Let $x_n \in X$ be an instance with $X = (x_1, x_2, x_3, ..., x_N)$ and $x_n = (a_1, a_2, a_3, ..., a_V)$ with attribute $a_v \in \mathcal{F}$, the feature space, with

$$a_v = \begin{cases} continuous, a_v \in \mathbb{R} \\ discrete, a_v \in \mathbb{N} \\ categorical, a_v \in \{0, 1\} \end{cases}$$

Be x_{n_m} a group member instance o_i of the mth group $\mathcal{G}_m$ with

$$\mathcal{G}_m = (o_1, o_2, o_3, ..., o_{N_m}) \tag{1}$$

and $\mathcal{D}_{GAD}$ a group anomaly detection dataset, which is a set $\mathcal{G}$ of all groups:

$$\mathcal{D}_{GAD} = (\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, ..., \mathcal{G}_M) \tag{2}$$

The objective of the group anomaly detection task is to distinguish normal in-distribution groups from abnormal out-of-distribution groups $\mathcal{G}_{\mathcal{A}}$ with the help of a pseudo group $\mathcal{G}^{(ref)}$ as an approximated reference for normal in-distribution groups. Therefore, a characterization function f with

$$f_\Theta : \mathbb{R}^{N_m \times V} \to \mathbb{R}^D \tag{3}$$

and an aggregation function g with

$$g_\phi : \mathbb{R}^D \to \mathbb{R}^D \tag{4}$$

compose to

$$\mathcal{G}^{(ref)} = g_\phi(f_\Theta(\mathcal{G})) \tag{5}$$

where f_Θ maps the groups $\mathcal{G}_m$ to D-dimensional feature vectors representing the relationship characteristics of its group members o_i and g_ϕ aggregates them to one D-dimensional feature vector representing one reference $\mathcal{G}^{(ref)}$ for the distribution of normal groups.

Finally, the abnormality of a group is defined by a group anomaly score y_{score} measuring the deviation by a distance measure $d(\cdot, \cdot) \geq 0$, between $\mathcal{G}_m$ and the normal group reference $\mathcal{G}^{(ref)}$. Thus, the abnormality score y_{score} is defined as follows:

$$y_{score} = d(\mathcal{G}^{(ref)}, \mathcal{G}_m) \tag{6}$$

whereby the decision between normal and abnormal groups is defined by a threshold γ with

$$y_{label} = \begin{cases} 1, & y_{score} \geq \gamma \\ 0, & otherwise \end{cases} \tag{7}$$

Having the group anomaly detection problem described according to [2], we also elaborate on the task of trajectory anomaly detection aligning with the notations of [3] with their slightly different problem of group trajectory anomaly detection instead of the here described individual trajectory anomaly detection:

A trajectory point p is defined as

$$p = (a_1, a_2, a_3, ..., a_V) \tag{8}$$

A trajectory point embedding e is defined by input embedding h_1 (cf. Figure I) for each point p as

$$e = h_1(p) \tag{9}$$

Since word sentences and trajectories can both be considered as sequences we create BERT-based embeddings [15] for trajectories by defining trajectory segments s_i with e.g. $s_1 = (e_1, e_2), s_2 = (e_3, e_4), ...$ for S segments of segment length $L_s = 2$, which represent local sequences within a trajectory. In addition to that, each segment s_i is mapped to a segment embedding h_2, which acts as an offset for each related trajectory point embedding e_n. The sum of both, segment member e_n and segment embedding h_2, is denoted as trajectory segment part sp with

$$sp_{i,n} = e_n + h_2(s_i) \tag{10}$$

A trajectory Λ is defined as

$$\Lambda_m = (sp_{11}, sp_{12}, sp_{23}, sp_{24}, ..., sp_{SN_m}) \tag{11}$$

A trajectory dataset $\mathcal{D}_{Traj}$ is defined as

$$\mathcal{D}_{Traj} = (\Lambda_1, \Lambda_2, \Lambda_3, ..., \Lambda_M) \tag{12}$$

By considering the task of detecting abnormal individual trajectories as group anomaly detection problem, the following associations are identified:

A trajectory Λ_m applies to the semantic of a group $\mathcal{G}_m$ by considering trajectory segments s in form of its segment parts sp as group members o. They are represented by point embeddings e adding a shared segment embedding h_2 as offset (cf. Eq. 10). The embeddings e represent trajectory points p and a trajectory point p_i is associated with an instance x_n.

Thus, individual abnormal trajectories can be detected similarly to the group anomaly detection problem (cf. Eq. 5 and Eq. 6) as follows:

$$\Lambda^{(ref)} = g_\phi(f_\Theta(\Lambda)) \tag{13}$$

$$\hat{y}_{score} = d(\Lambda^{(ref)}, \Lambda_m) \tag{14}$$

After revealing the associations between the GAD approach of [2] and our approach for trajectories, also our proposed GADFormer approach (cf. Section III) shows the potential to be trained for each arbitrary group anomaly detection problem on sequences or non-ordered sets, as far as a group to member relationship exists.

III. GADFormer

In this section we propose GADFormer, a deep BERT based transformer encoder model architecture for attention-based Group Anomaly Detection (GAD). After we showed in Section II by the example of [2] theoretically that the GAD problem can also be applied to trajectory coordinates, we introduce in this section with GADFormer a new deep GAD model and demonstrate its performance on trajectory datasets in Section IV. Figure 2 provides an overview to its model architecture in combination with examples of 2D trajectory point inputs, but also high-dimensional group members (trajectory points) are possible.

A. Architecture and Loss Objective

Differently to the GAD characterization and aggregation function (cf. Eq. 3 and Eq. 4) of the deep generative models of [2] our deep GADFormer Ψ models the characterization and aggregation functions as follows with

$$\Psi : g_\Phi(f_\theta(\Lambda_m)) \rightarrow \hat{p}_m. \tag{15}$$

The characterization function f_θ of GADFormer maps the bidirectional relationships between group members o_i of a group $\mathcal{G}_m$ (representations of segment parts sp of a trajectory Λ_m) to a multi-head self-attention-weight feature map $b_{\mathcal{G}_m}$, representing the behavior of an individual group (an individual trajectory path pattern). This is realized by a BERT [15] encoder, a composition of layers (h_1 and h_2) for input embedding, positional encoding and multi-head self-attention blocks (cf. Figure 2) using group member embeddings pe as input tokens. In order to extend the possible value range for an improved feature extraction, we replace the ReLU activation function of the standard FFN of [16] with Tanh.

$$b_{\Lambda_m} = f_\theta(\Lambda_m) \tag{16}$$

The aggregation function g_Φ of GADFormer approximates instead of a distribution for normal group representations $\mathcal{G}^{(ref)}$ with distance measure d a probability p_m for abnormal group behavior (abnormal trajectory path pattern). This is realized by non-linear layer blocks (2 linear projections with ReLU and a final output layer with linear compression and Sigmoid non-linearity) as part of the task output block g_Φ, which maps the group behavior characteristics b_{Λ_m} to a task specific feature map representation z_{Λ_m}.

$$z_{\Lambda_m} = g_\Phi(b_{\Lambda_m}) \tag{17}$$

After compression, the sigmoid function maps representation z_{Λ_m} to a probability $\hat{p}_m$ for group abnormality, with $\hat{p}_m \in [0, 0.5]$ for normal groups and $\hat{p}_m \in]0.5, 1]$ for abnormal groups (trajectories).

$$\hat{p}_m = \sigma(z_{\Lambda_m}) \tag{18}$$

Because of the rare label availability for trajectories, the loss objective of GADFormer is defined for an unsupervised and semi-supervised learning setting assuming the majority of instances to be normal. Therefore, we define the binary cross entropy loss $\mathcal{L}_{BCE}$ as our loss function (cf. Eq. 20). We consider this loss function as a suitable choice, since entropy $H(\hat{p}_m)$ as a measure of unpredictability is $H(\hat{p}_m) = 1$ when the model is most uncertain about its abnormality prediction, and $H(\hat{p}_m) = 0$ when the model is very certain about its abnormality prediction.

$$H(\hat{p}_m) = \begin{cases} 1, & \hat{p}_m = 0.5 \\ 0, & \hat{p}_m = 0 \wedge \hat{p}_m = 1 \\ ]0, 1[, & otherwise \end{cases} \tag{19}$$

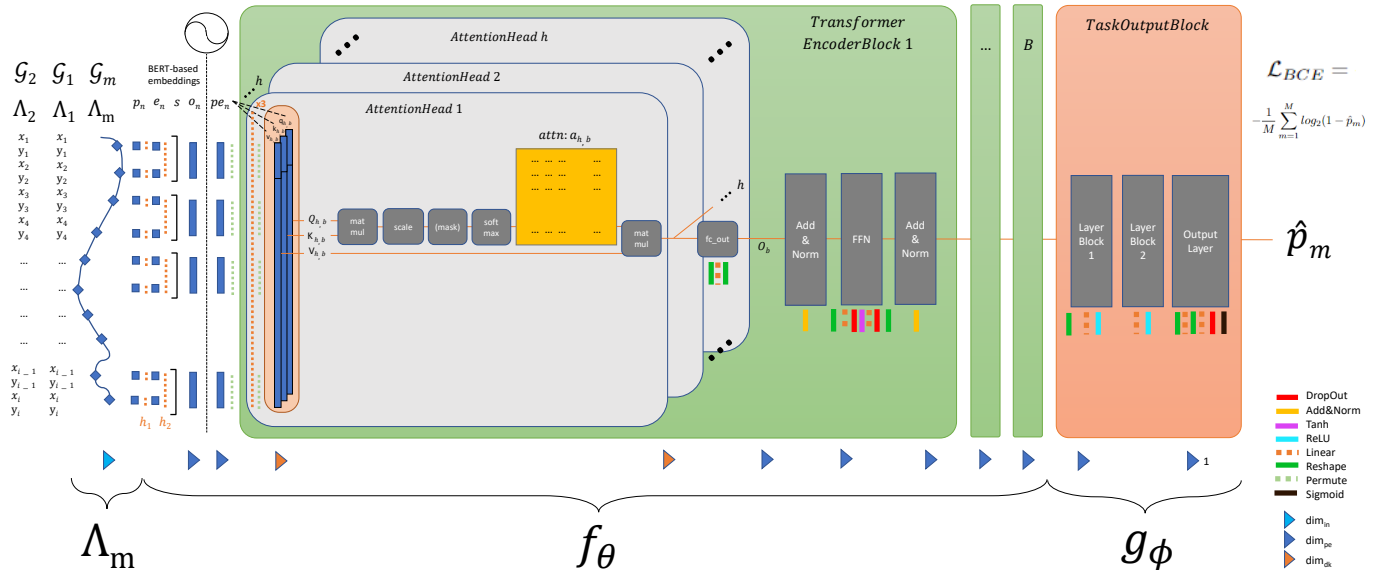

Fig. 2. GADFormer architecture overview.

Due to the heavily imbalanced learning setting with a large majority of normal groups (trajectories) one can neglect the minority of abnormal groups (trajectories) and set a fix auxiliary target probability $p_m = 0$ for certain normal-predictions ($H(\hat{p}_m) = 0$) for the majority of normal group probabilities $\hat{p}_m$. In case the model faces true abnormal groups, then it is rather uncertain about its decision yielding a probability close to $\hat{p}_m = 0.5$ resulting in a high entropy loss, whereas true normal groups, on whose pattern the model is trained, result in a low entropy loss for $\hat{p}_m \approx 0$.

$$\mathcal{L}_{BCE} = \frac{1}{M}\sum_{m=1}^{M} p_m log_b(\hat{p}_m) + (1-p_m) log_b(1-\hat{p}_m)$$
$$\overset{p_m=0}{\Longrightarrow}$$
$$\mathcal{L}_{BCE} = \frac{1}{M}\sum_{m=1}^{M} log_b(1-\hat{p}_m) \tag{20}$$

Since in our setting the model effectively predicts only abnormality probabilities for the range of normal groups with $\hat{p}_m \in [0, 0.5]$, where $\hat{p}_m = 0$ means that a group (trajectory) is not abnormal at all, the abnormality of a group is defined by a group anomaly score $\hat{y}_{score} = \hat{p}_m$ for our GADFormer approach.

B. Training

Anomalous trajectories are rare by definition and labeling by domain experts tends to be rather expensive. We address this challenge by two different learning settings which are: 1) Unsupervised learning, which requires no labels at all under the assumption that the ratio of anomalous trajectories is low and has no remarkable influence during model training. 2) Semi-Supervised Learning, which relies on verified normal samples only. As proposed in the section before, these learning settings allow us to set a fix auxiliary target probability $p_m = 0$, so that no ground truth for abnormal trajectories is needed for the GADFormer training. In order to let f_θ learn expressive representations for g_Φ we start the training with frozen task layers, which get unfrozen as soon as validation loss stops decreasing. Furthermore, we use early stopping, learning rate scheduling ReduceOnPlateau and RAdam for optimization. Please see our supplementary material[1] for further details.

C. Model Transparency

Deep learning models are known to be rather complex and their training usually requires a deep understanding for its model architecture, losses, preprocessing and data distributions to take the right decisions for fine-tuning, but still then it partially remains a blackbox as more layers, blocks and parameters exist.

In order to achieve a higher model transparency addressing *CH6*, one of the main deep anomaly detection challenges of [18], we introduce a Block Attention-anomaly Score (BAS) for our GADFormer model. BAS can be seen as a further interpretable explanator for Model Inspection, solving a so called Open-The-Box-Problem[19], which allows to indicate how each layer of the transformer encoder model contributes to distinguish inputs of different ground truth classes. Class-overlapping scores in the final layer are a potential indicator for false positives and negatives respectively. Hence, BAS enables for model inspection with the goal of identifying optimization potential in the model architecture using attention matrix scores deviating from the attention matrix score mean without plausible correlation to its ground truth and neighboring ground truths. BAS follows the assumption that

[1]https://github.com/lohrera/gadformer

in case of the aggregated attention of a group of layer heads is anomalous then also the model input, in our case the group member instances of a trajectory, is anomalous. Considering the example of Figure 3 for a good model performance, this assumption holds for the majority of abnormal inputs across nearly all layers, especially for the final layer in which the amount of false positives decreases.

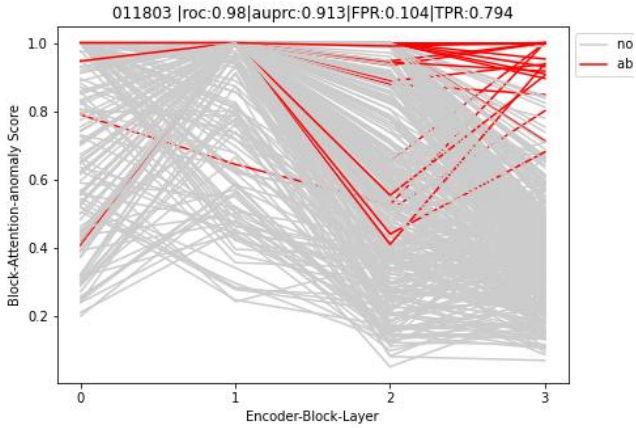

Fig. 3. BAS in case of good model performance.

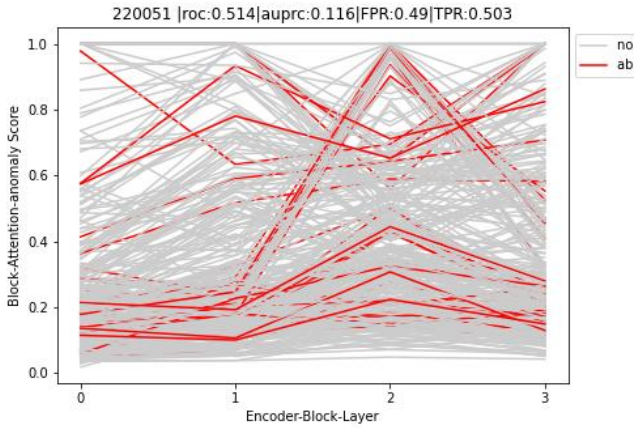

Fig. 4. BAS in case of bad model performance.

BAS represents in a transformer encoder block layer a multi-head-attention group anomaly by the ratio between distance of an aggregated block attention matrix $a_{m,b}$ and its normal block average a_b and distance of a_b to the average of $topN$ abnormal aggregated attention matrices $a_{topN,b}$ (cf. line 10 of Algorithm 1). This allows to show the capability of a transformer encoder block layer to 1) generally distinguish pattern of different groups (trajectories) and 2) to separate between normal and abnormal groups of attention head weights for individual trajectories (groups). Therewith BAS is different to the work of [20], which aims to identify single feature-relevant attention heads by maximum attention weights and histograms instead of using average attention distances.

Since the cells of an attention matrix $a_{m,h,b}$ contain the scaled dot product of $q_{m,h}$ and $k_{m,h}$ (two projected group embeddings projected from input tokens pe, cf. Figure 2), their similarity weights the importance of the bidirectional relationship of a group member pair in different heads h and with that, focuses with different views to the behavioral pattern of a group (trajectory path pattern in the context of the task of GAD on individual trajectories), whereas its concatenated projected dot product with the third projected group member embeddings $v_{m,h}$ provides the overall-importance-weighted attention output matrix $O_{m,b}$ emphasizing task relevant pattern of a concrete group m (task relevant trajectory path segments of an individual trajectory Λ_m).

After emphasizing the role of the attention mechanism, we describe in Algorithm 1 how to calculate BAS in detail. The inputs of this algorithm are the attention matrices a of the transformer encoder blocks and the euclidean distance measure. Further parameters are the ratio for the top N abnormal groups, block index b and the amount of groups M.

Algorithm 1 BAS Algorithm Pseudo Code

Input: group attention matrices a, distance measure d
Parameters: $ratio_{topN} = 0.05$, block b, groups M
Output: block attention-anomaly scores bas

1: $a_{m,b} = \mu(a_{m,h,b})$
2: **if** training **then**
3: $\quad tmp\_a_b = \mu(a_{m,b})$
4: $\quad topN = \lceil ratio_{topN} * M \rceil$.
5: $\quad idx\_a_b = rank(d(a_{m,b}, tmp\_a_b), topN, dsc)$
6: $\quad idx\_n_b = idx\_all \setminus idx\_a_b$
7: $\quad global \quad a_{topN,b} = \mu(a_{m,b}[idx\_a_b, :, :])$
8: $\quad global \quad a_b = \mu(a_{m,b}[idx\_n_b, :, :])$
9: **end if**
10: $bas_{m,b} = min(1., \frac{d(a_{m,b}, a_b)}{d(a_{topN,b}, a_b)})$
11: **return** bas

The average attention matrix $a_{m,b}$ for a group m in heads h is calculated for all attention matrices $a_{m,h,b}$ (line 1). During training also their temporary average tmp_a_b over all groups is calculated (line 3) to obtain their $topN$ most distant abnormal group indices idx_a_b (line 5), whose difference to all indices result in the remaining normal group indices idx_n_b (line 6). Based on these indices a global average attention heads mean for normal (a_b) and abnormal ($a_{topN,b}$) head attention averages is calculated (line 7-8) during training in order to have a solid normal and abnormal representation for distance calculation. Next, the distance between a group attention matrix $a_{m,b}$ to the normal group attention matrix average a_b as well as the distance between the normal group attention matrix average a_b and the abnormal group attention matrix average $a_{topN,b}$ is used to request the ratio between both which represents the Block Attention-anomaly Score $bas_{m,b}$ (line 10-11). Figure 3 shows, that the first encoder block layers (0-2) are not able to distinguish between normal and abnormal trajectories whereas in the last layer the amount of potential false positive scores gets less indicating a better capability of the model to attend to features of abnormal trajectories. The BAS within the layers of Figure 4 indicate that the model is over all layers not really able to distinguish between normal and abnormal trajectory scores, not even between the characteristics of single trajectories within the class of normal trajectories.

In summary, the BAS provides us a view to the transformer encoder block layers for model inspection and allows us to reason reasonable changes to hyperparameters and model architecture to improve the performance of the model. Providing a further answer to "Do Transformer Attention Heads Provide Transparency in Abstractive Summarization?"[20], we are able to provide attention block transparency in terms of which degree the averaged attention of a group of self-attention-heads within one layer is normal or abnormal.

IV. Experiments

In this section we evaluate the performance of our GADFormer approach on synthetic and real-world datasets and compare it against related works like GRU and MainTulGAD, whose approaches are state of the art methods for individual trajectory anomaly detection. For details related to datasets, architectures, hyperparameters, results and code see our supplementary material[1].

A. Experimental Setup and Datasets

For our experiments we tested our approach on synthetic data[1] and three real-world datasets, i.e., amazon driving routes[2], Deutsche Bahn cargo container routes[3] and brightkite checkin routes [4] (cf. Table I). The synthetically generated trajectory dataset consists of trajectory steps, one per row, where each has an id, sequence step, xcoord, ycoord, and a label whether its trajectory (=group) is anomalous (1) or normal (0).

All hyperparameters are empirically selected by grid search based on validation loss convergence and additionally validated by model inspection with our proposed block attention anomaly score (cf. BAS Algorithm 1) in order to find ideal training parameters and a model architecture avoiding overfitting or insufficient model complexity.

The code for GADFormer is implemented in Python utilizing PyTorch. For best possible comparison, the architecture for GRU is identical to GADFormer except using GRU-layers instead of encoder block attention layers and only input embedding (cf. Eq. 9) instead of BERT segmentation. Our MainTul [17] version (MTGAD) uses kNN-trajectory-augmentations and a student-teacher-architecture for feature extraction like the original, but adapts to sequential trajectory coordinates instead of time-dependent categorical checkins. These approaches represent the technically most related work for a fair comparison. Extended results with less related traditional methods can be found in supplementary material[1].

[2]https://github.com/amazon-science/goal-gps-ordered-activity-labels

[3]https://data.deutschebahn.com/dataset/data-sensordaten-schenker-seefrachtcontainer.html

[4]https://snap.stanford.edu/data/loc-brightkite.html

TABLE I
Dataset Overview.

dataset	setting	all	normal	abnormal	trajLen
synthetic[1]	unsup	3400	3083	317	72
synthetic[1]	semi	3400	3271	129	72
amazon[2]	unsup	805	760	45	72
amazon[2]	semi	776	760	16	72
dbcargo[3]	unsup	272	229	43	72
dbcargo[3]	semi	245	229	16	72
brightkite[4]	unsup	2241	2033	208	500
brightkite[4]	semi	2108	2033	75	500

B. Evaluation

For the evaluation of our model, we follow the goals of having a low miss rate (false negatives) as well as achieve as less false alerts (false positives) as possible. In addition to that, the quality of the model scores needs to be evaluated. Therefore and to be comparable to related approaches, we evaluate the model performance by AUROC and AUPRC.

- $TPR = \frac{TP}{P}; FPR = \frac{FP}{N}; FNR = 1 - TPR$
- Precision = $\frac{TP}{TP+FP}$; Recall = $\frac{TP}{TP+FN}$
- AUPRC = AP = Precision vs. Recall
- AUROC = TPR vs. FPR

C. Results and Discussion

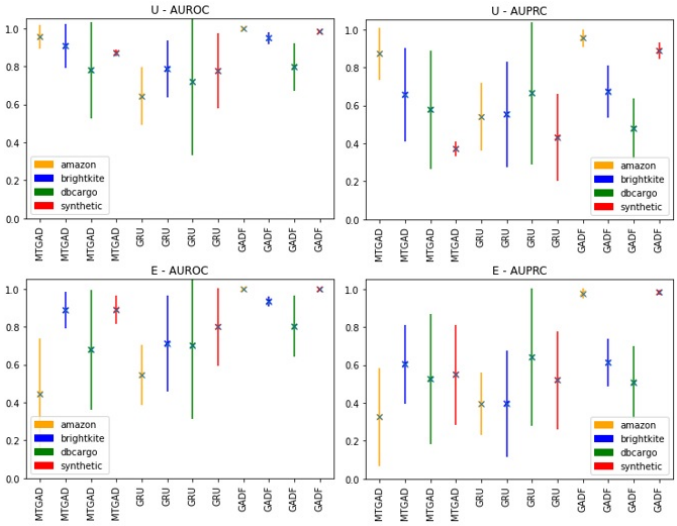

Fig. 5. Robustness results for synthetic and real world datasets of Table II. Span shows the stddev of 10 seeds; cross is the used mean performance.

Our approach GADFormer (GADF) outperforms related works GRU and MainTulGAD (MTGAD) on all synthetic and real world datasets in terms of AUROC and AUPRC except for AUPRC on dbcargo for which an approach like GRU could achieve a better performance for both un- and

TABLE II
Results on synthetic and real world datasets for (U)nsupervised- and s(E)mi-supervised setting (cf. Figure 5).

	dataset	amazon		brightkite		dbcargo		synthetic	
		auroc	auprc	auroc	auprc	auroc	auprc	auroc	auprc
U	GRU	0.642	0.539	0.786	0.552	0.718	**0.664**	0.775	*0.431*
	MTGAD	*0.956*	*0.872*	*0.907*	*0.656*	*0.779*	*0.577*	*0.87*	0.371
	GADF	**0.997**	**0.955**	**0.948**	**0.672**	**0.797**	0.478	**0.982**	**0.887**
E	GRU	*0.545*	*0.394*	0.711	0.396	*0.701*	**0.64**	0.799	0.52
	MTGAD	0.445	0.325	*0.887*	*0.604*	0.678	*0.526*	*0.889*	*0.549*
	GADF	**0.998**	**0.976**	**0.933**	**0.612**	**0.801**	0.507	**0.997**	**0.982**

TABLE III
RESULTS ON SYNTHETIC DATASET WITH NOISE ABLATIONS FOR (U)NSUPERVISED AND S(E)MI-SUPERVISED SETTING

	exp	noise .0		noise .2		noise .5	
		auroc	auprc	auroc	auprc	auroc	auprc
U	GRU	0.766	*0.514*	0.731	*0.383*	0.626	*0.165*
	MTGAD	*0.869*	0.376	*0.822*	0.256	*0.717*	0.149
	GADF	**0.97**	**0.892**	**0.949**	**0.831**	**0.863**	**0.537**
E	GRU	0.788	0.585	0.759	0.479	0.665	0.223
	MTGAD	*0.952*	*0.766*	*0.89*	*0.547*	*0.792*	*0.316*
	GADF	**0.989**	**0.95**	**0.98**	**0.919**	**0.944**	**0.803**

TABLE IV
RESULTS ON SYNTHETIC DATASET WITH NOVELTY ABLATIONS FOR (U)NSUPERVISED AND S(E)MI-SUPERVISED SETTING.

	exp	novelty .0		novelty .01		novelty .05	
		auroc	auprc	auroc	auprc	auroc	auprc
U	GRU	0.766	*0.514*	0.832	0.585	0.818	0.496
	MTGAD	*0.882*	0.42	*0.935*	*0.588*	*0.923*	*0.504*
	GADF	**0.97**	**0.892**	**0.978**	**0.865**	**0.969**	**0.726**
E	GRU	0.788	0.585	0.849	0.652	0.841	0.574
	MTGAD	*0.964*	*0.802*	*0.977*	*0.867*	*0.97*	*0.797*
	GADF	**0.989**	**0.95**	**0.986**	**0.921**	**0.986**	**0.841**

semisupervised settings. Considering Figure 5 and Table II, GADFormer demonstrates its stability across all datasets with lowest standard deviations over 10 seeds. For datasets amazon, brightkite and synthetic its AUROC standard deviation is close to zero. AUROC performances over 0.8 on real-world datasets in semisupervised settings highlight its relevance for real-world-domains, especially for amazon routes for which it achieved performances over 0.95 for all metrics. Also on brightkite (a dataset with long sequences of 500 steps) showed our transformer-based approach still the best performance, demonstrating its superiority against GRU and MTGAD which both are at least partially based on recurrent neural networks. MTGAD with its self-supervised augmented kNN-trajectories and its combined student-teacher-approach of LSTM and multi-head attention shows performances close to but slightly weaker than GADFormer except for AUPRC of dbcargo. In ablation studies for detecting noise-distorted and novel anomalies shows GADFormer a comparable strong performance even for high noise and novelty ratios from 0 up to 0.5 or 0.05 respectively (cf. Table III and Table IV). Summarizing, compared to related work, GADFormer (GADF) can be considered as a robust approach, but despite its strong false and miss alert rates (evaluated via AUROC and AUPRC) it depends on the domains if these performances are sufficient.

V. RELATED WORK

Reviewing the literature for most related approaches, we could identify the following related work, which gets distinguished from our approach within this section. Instead of considering the detection of individual trajectory anomalies as a Group Anomaly Detection problem as our approach does, the vision in the works of [21, 22] is to observe trajectories as a NLP problem. They map trajectory coordinates to hexagon-based hexadecimal-words as input for pretrained BERT models for several tasks, but do not provide a concrete model architecture for group anomaly detection based on projected trajectory segments as BERT-based embeddings. Another work of [17] uses for the task of Trajectory-User-Linking (TUL), instead of GAD, a combination out of RNN and transformer network with cross entropy loss but compared to our approach, they do not take trajectory coordinates and segments into account and the model lacks in layer transparency. Addressing long-range trajectory anomaly detection as well the work of [23] proposes an unsupervised normalizing flow (NF) model. They utilize trajectory segments and negative log-likelihood as well but use it in combination with NF-based density estimation. The work of [24] introduced the problem of group trajectory outlier detection (GTOD), which is also addressed by [3], and provide the approach CDkNN, which creates DBSCAN-based microclusters, pruned by kNN and scored with a specific pattern mining algorithm. However, both works perform anomaly detection while considering complete individual trajectories as group members, whereas we address the slightly different problem of considering single trajectory points as group members for the problem group anomaly detection. The work of [25] proposes a model for content-aware anomaly detection on event log messages instead of anomalous trajectories as our approach. Their approach takes additionally the content of the messages into account and allows to run it, as our approach, by the task-specific encoder-part or, differently as ours, by the typical BERT[15] encoder-decoder architecture. Summarizing the identified related work, there is best to our knowledge no transparent attention-based transformer-encoder-approach for group anomaly detection on coordinates-based trajectories.

VI. CONCLUSION

In this work we proposed GADFormer, a transformer-encoder-architecture, capable to perform attention-based group anomaly detection in an unsupervised and semi-supervised setting. We emphasized, how the detection of individual anomalous trajectories can be solved as a Group Anomaly Detection (GAD) problem for BERT based transformer models. Furthermore, we introduced BAS, a Block Attention-anomaly Score to allow model inspection for transformer encoder blocks for the task of GAD and improve with that its transparency in terms of answering to which degree the attention of the group of self-attention-heads is normal or abnormal. Extensive ablation and robustness studies addressing trajectory noise and novelties on synthetic and real world datasets demonstrated, that our approach is on par with related attention-based approaches like GRU. Further potential for improvement could be to approximate a normal-group-distribution instead of abnormal-group-probabilities by the output-block of our model, combining the attention-based group pattern extraction of our approach and the group anomaly scoring and loss objectives of [2]. Vice versa, with appropriate preprocessing, the performance of the GADFormer model architecture could also be evaluated on image data, audio or text data. Moreover, the reliability of the probabilities of our approach could be further investigated

according to the work of [26] as well as the relevance of single group member instances for a specific model prediction.

References

[1] Liang Xiong, Barnabás Póczos, and Jeff Schneider. "Group Anomaly Detection Using Flexible Genre Models". In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., 2011, pp. 1071–1079.

[2] Raghavendra Chalapathy, Edward Toth, and Sanjay Chawla. "Group Anomaly Detection Using Deep Generative Models". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I*. Vol. 11051. Springer, 2018, pp. 173–189.

[3] Asma Belhadi et al. "Machine Learning for Identifying Group Trajectory Outliers". In: *ACM Trans. Manage. Inf. Syst.* 12.2 (Jan. 2021).

[4] Andreas Lohrer, Johannes Josef Binder, and Peer Kröger. "Group Anomaly Detection for Spatio-Temporal Collective Behaviour Scenarios in Smart Cities". In: *Proceedings of the 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. IWCTS '22. Seattle, Washington: Association for Computing Machinery, 2022.

[5] Krikamol Muandet and Bernhard Schölkopf. "One-Class Support Measure Machines for Group Anomaly Detection". In: UAI'13. Bellevue, WA: AUAI Press, 2013, pp. 449–458.

[6] Benjamin Nachman and David Shih. "Anomaly detection with density estimation". In: *Phys. Rev. D* 101 (7 Apr. 2020), p. 075042.

[7] Rose Yu, Xinran He, and Yan Liu. "GLAD: Group anomaly detection in social media analysis". In: 2014.

[8] Ralph Foorthuis. "On the nature and types of anomalies: a review of deviations in data". In: *International Journal of Data Science and Analytics* 12 (2020), pp. 297–331.

[9] Zhongqiu Wang et al. "Unsupervised learning trajectory anomaly detection algorithm based on deep representation". In: *International Journal of Distributed Sensor Networks* 16 (2020).

[10] Xiang Jiang et al. "Improving point-based AIS trajectory classification with partition-wise gated recurrent units". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 4044–4051.

[11] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780.

[12] Kyunghyun Cho et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014.

[13] Liang Xiong et al. "Hierarchical Probabilistic Models for Group Anomaly Detection". In: *International Conference on Artificial Intelligence and Statistics*. 2011.

[14] Bernhard Schölkopf et al. "Estimating the Support of a High-Dimensional Distribution". In: *Neural Computation* 13.7 (2001), pp. 1443–1471.

[15] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[16] Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[17] Wei Chen et al. "Mutual Distillation Learning Network for Trajectory-User Linking". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. International Joint Conferences on Artificial Intelligence Organization, July 2022, pp. 1973–1979.

[18] Guansong Pang et al. "Deep Learning for Anomaly Detection: A Review". In: *ACM Comput. Surv.* 54.2 (Mar. 2021).

[19] Riccardo Guidotti et al. "A Survey of Methods for Explaining Black Box Models". In: *ACM Comput. Surv.* 51.5 (Aug. 2018).

[20] Joris Baan et al. "Do Transformer Attention Heads Provide Transparency in Abstractive Summarization?" In: *ArXiv* abs/1907.00570 (2019).

[21] Mashaal Musleh, Mohamed F Mokbel, and Sofiane Abbar. "Let's speak trajectories". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 2022, pp. 1–4.

[22] Mashaal Musleh. "Towards a unified deep model for trajectory analysis". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 2022, pp. 1–2.

[23] Madson L. D. Dias et al. "Anomaly Detection in Trajectory Data with Normalizing Flows". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8.

[24] Youcef Djenouri et al. "Fast and Accurate Group Outlier Detection for Trajectory Data". In: *New Trends in Databases and Information Systems*. Cham: Springer International Publishing, 2020, pp. 60–70.

[25] Shengming Zhang et al. "CAT: Beyond Efficient Transformer for Content-Aware Anomaly Detection in Event Sequences". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 4541–4550.

[26] Zhengbao Jiang et al. "How Can We Know What Language Models Know?" In: *Transactions of the Association for Computational Linguistics* 8 (2019), pp. 423–438.