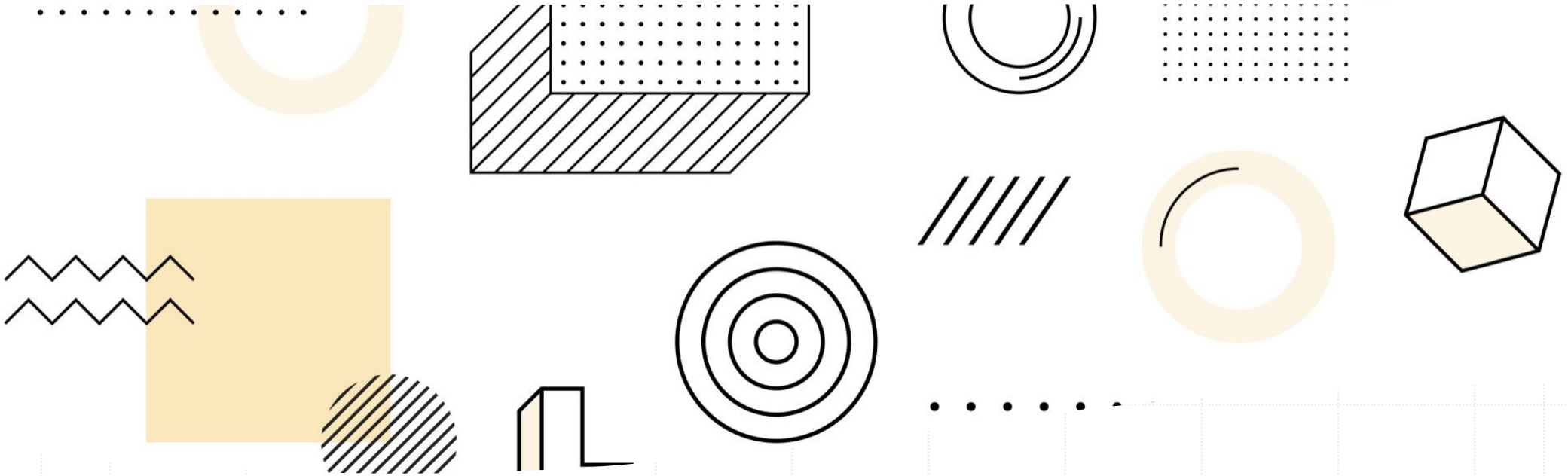


ENCODiT

Error flagging and Neutralization using Conformal Out-of-Distribution Detection in Time-Series Data

Group 14

1. Kartik Anant Kulkarni (210493)
2. Rishi Agarwal (210849)
3. Emaad Ahmed (210369)
4. Dhruva Singh Sachan (210343)

Acknowledgement

We would like to thank the authors **Ramneet Kaur, Kaustubh Sridhar, Sangdon Park, Susmit Jha, Arinban Roy, Oleg Solosky,** and **Insup Lee** for this fantastic piece of research which enabled us to learn about the practical problem that Out of Order Detection and allowed us to tinker with their novel algorithm using our ideas. We also thank the course Instructor, **Prof. Indranil Saha** for providing us the opportunity to work on this interesting bit of research as a course project, which really helped us get a broader outlook of the extensive usage of CPS and Embedded systems and the broad research actively taking place around it. We also thank our TA, **Mr. Vaibhav Tanwar**, for his support whenever we needed it.

Outline

Paper Review

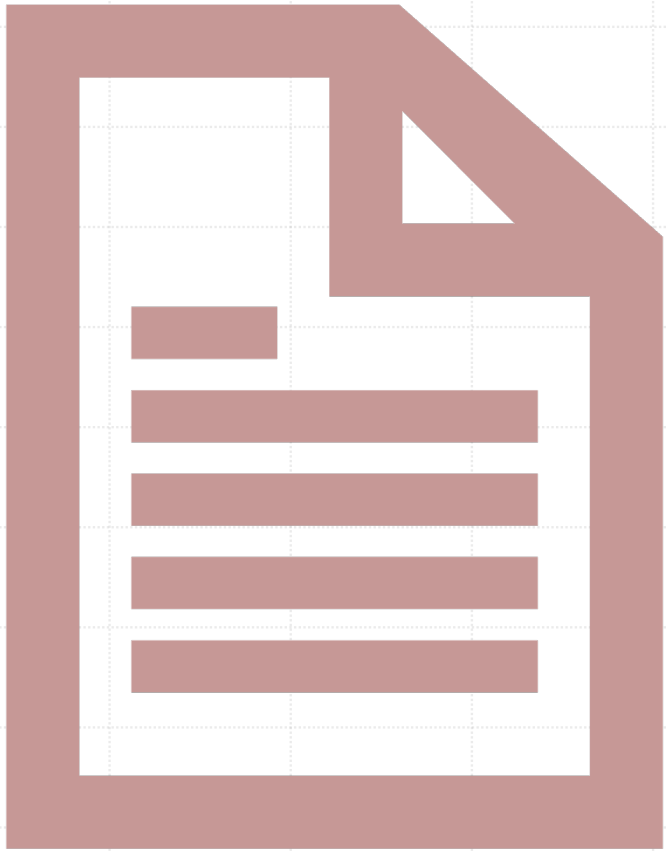
- Problem Statement
- Approaches and Demerits
- Mathematical Formulations
- Results
- General Architecture

Verification and Experimentation

- Robocup MSL
- Dynamic Window Size
- RNN Based Model
- Future Work
- Summary

Contribution

Paper Review



Problem Statement

Usage of Learning Enabled Components (LECs) in Cyber-Physical Systems (CPS)

What is Out Of Distribution (OOD)?

Safety Insurance in Real-time Applications

Common Solutions by Cascading Models

Demerits of Other Approaches

- **Temporal Relationships** in Time-Series Sequences not exploited. E.g. Frozen Camera, Drift Detection etc. cannot be detected by most point-based methods.
- No work has been done on **non-pictorial** data. E.g. GAIT Detection in medical CPS, etc. has not explored.
- Many approaches cannot provide a **bound on the error rate** making them unreliable.

Frozen Camera

Drift

Key Contributions of the Paper

Novel Measure for OOD Detection in Time-Series Data for CPS

- Proposed a **non-conformity measure** that is defined on the window containing information about the sequence of time-series datapoints for enhancing detection of the temporal OODs
- Used a model trained to learn **iD temporal equivariance** via the auxiliary task of **predicting an applied transformation** on windows drawn from the training distribution of LEC
- Used the prediction error as the non-conformity measure in **ICAD** framework for OOD detection in high-dimensional time-series input data

Enhanced detection performance

- Use **Fisher's method** as an ensemble approach for combining predictions from multiple conformal detectors based on the proposed measure

CODiT

- Compute n **independent** p**-values** of the input from the proposed measure in the ICAD framework as an OOD window is less likely to behave as iD under multiple transformations.
- Combine these values by Fisher's method leads to the proposed detector CODiT with a **bounded false alarm rate.**
- Algorithm works on any general time series data providing SoTA results

Table 1: Capabilities of detectors in time-series data for CPS.

OOD Detector	False Alarm Rate Guarantees	Temporal OODs	Non-vision Data
VAE [5]	✓	?	✓
β-VAE[27]	✓	?	✓
Memory [36]	✗	?	✓
Feng et al.'s [10]	✗	✓	✗
CODiT (Ours)	✓	✓	✓

Note: "?" is used as there is no clear mention of application to time series data for temporal OODs in the papers.

Comparison With Other Approaches

- It is unclear how to directly apply individual point detectors to time-series data with the "ICAD guarantees", because:
  - Even if we apply these detectors to individual datapoints in the time-series window independently, we do not know how to **combine detection verdicts** on these datapoints for detection on the window.
  - For detection guarantees by ICAD, it is required that all non-conformity scores for p-value computation to be **IID**.
- CODiT's approach is **not point-based** and is **not limited to** CNNs. It is **not image-specific** as it does not incorporate any sophistications like optical flows. Further, it is also **computationally efficient**.

Mathematical Formulation

Definitions

Definition 1 (Schmidt and Roth, 2012). *For a set X, a function f is defined to be equivariant with respect to a set of transformations G, if there exists the following relationship between any transformation $g \in G$ of the function's input and the corresponding transformation g' of the function's output:*

$$f(g(x)) = g'(f(x)), \forall x \in X. \tag{1}$$

Definition 2 (Jenni and Jin, 2021). *Temporal equivariance of a function f from equation 1 is defined on a set X of windows of consecutive time-series datapoints and with respect to a set G of temporal transformations.*

ICAD

The training dataset X of size l is split into a *proper training set* $X_{\text{tr}} = \{x_j : j = 1, \ldots, m\}$ and a *calibration set* $X_{\text{cal}} = \{x_j : j = m+1, \ldots, l\}$. Proper training set X_{tr} is used in defining NCM. In the example of reconstruction error by a VAE as the non-conformity score, the VAE trained on X_{tr} is used for computing the error. Calibration set X_{cal} is a held-out training set that is used for computing p-value of an input. p-value of an input x is computed by comparing its non-conformity score $\alpha(x)$ with these scores on the calibration datapoints:

$$p\text{-}value(x) = \frac{|\{j = m+1, \ldots, l : \alpha(x) \leq \alpha(x_j)\}| + 1}{l - m + 1}. \tag{2}$$

Definitions (Contd..)

Fisher's Method

The same hypothesis can be tested by multiple conformal predictors and an ensemble approach for combining these predictions can be used to improve upon the performance of individual predictors. Fisher's method is one of these approaches for combining multiple conformal predictions or p-values of an input from (2). Fisher value of an input x from n p-values is computed as follows:

$$\text{fisher-value}(x) = r \sum_{i=0}^{n-1} \frac{(-\log r)^i}{i!}, \text{ where } r = \prod_{k=1}^{n} p_k. \qquad (3)$$

Lemma 2 (Toccaceli and Gammerman, 2017). *If n p-values, $p_1, \ldots, p_n$, are independently drawn from a uniform distribution of these values, then $-2\sum_{i=1}^{n} \log p_i$ follows a chi-square distribution with $2n$ degrees of freedom. Thus, the combined p-value is*

$$Pr\left(y \leq -2 \sum_{i=1}^{n} \log p_i\right) = r \sum_{i=0}^{n-1} \frac{(-\log r)^i}{i!},$$

where $r = \prod_{k=1}^{n} p_k$, y is a random variable following a chi-square distribution with $2n$ degrees of freedom, and the probability is taken over y. Moreover, the combined p-value follows the uniform distribution.

Algorithm with Guarantee on Correctness

Algorithm 1 CODiT: OOD Detection in Time-Series Data for CPS

Input: a window $X_{t,w}$ of time-series data, VAE model M trained on proper training set of the iD windows for LEC, distribution Q_{G_T} over the set G_T of temporal transformations, prediction error function f, n sets of calibration set alphas $\{\alpha_j^k : 1 \leq k \leq n, m+1 \leq j \leq l\}$, and desired false alarm rate $\epsilon \in (0,1)$

Output: "1" if $X_{t,w}$ is detected as OOD; "0" otherwise

for $k \leftarrow 1, \ldots, n$ **do**

  $g \sim Q_{G_T}$

  $\hat{g} \leftarrow M(g(X_{t,w}))$

  $\alpha \leftarrow f(g, \hat{g})$

  $p_k \leftarrow \frac{|\{j=m+1,\ldots,l:\ \alpha \leq \alpha_j^k\}|+1}{l-m+1}$

end for

$r \leftarrow \prod_{k=1}^{n} p_k$

if $r \sum_{i=0}^{n-1} \frac{(-\log r)^i}{i!} < \epsilon$ **then return** 1 **else return** 0

False Detection Rate Guarantees

- The false OOD detection on an input drawn from the training distribution is upper bounded by the specified detection threshold e.
- Proof of boundedness:
  - Input $\boldsymbol{X}$ and the Calibration Datapoints $\boldsymbol{X_{m+1}, \ldots, X_l}$ are independent and identically distributed (IID).
  - For any NCM trained on set $\boldsymbol{X_{tr}}$, p-values described in ICAD are uniformly distributed over {1/(l − m + 1), 2/(l − m + 1), . . . , 1}.
  - Thus, probability of misclassifying $\boldsymbol{X}$ as anomalous $\boldsymbol{P}$(p-value(x) < e) =

$$\sum_{i \leqslant i \leqslant (l-m-1)e} \frac{1}{l-m-1} = \frac{(l-m-1)e}{l-m-1} \leqslant e$$

Proposed NCM

- We use time-dependency between datapoints in a time-series window for detection, using deviation from the expected iD G_T -equivariance learned by a model on windows drawn from the training distribution as an NCM in ICAD for OOD detection in time-series data.
- G_T-equivariance is learned via an auxiliary task of predicting the applied temporal transformation on the window sampled from iD set.
- Error in the prediction of the applied temporal transformation on an input window $X_{t,w}$ is used as the NCM: PredictionError(g,M(g(Xt,w))). We used **CrossEntropyLoss**.

$$\alpha_i(X_{t,w}) = \text{PredictionError}(g_i, M(g_i(X_{t,w}))) \ : \ 1 \leq i \leq n, g_i \sim Q_{G_T}$$

Results

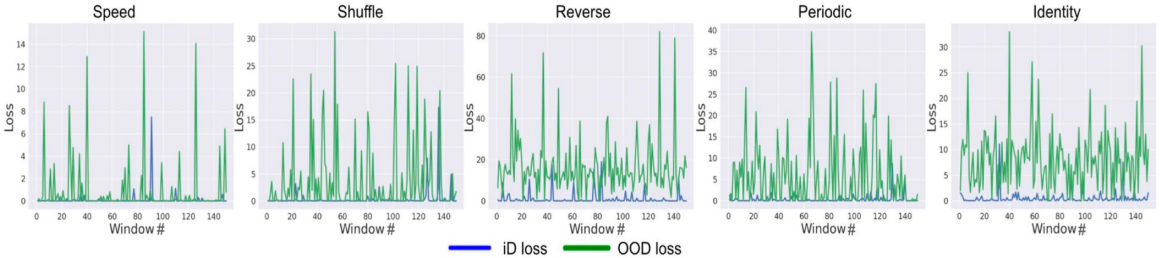


Figure 4: Higher values of TTPE-NCM = CrossEntropyLoss$(g, M(g(X_{t,w})))$ on OOD windows than on the test iD windows of the drift dataset. This shows that G_T-equivariance learned on the windows drawn from the training distribution of LEC is less likely to generalize on the windows drawn from OOD.

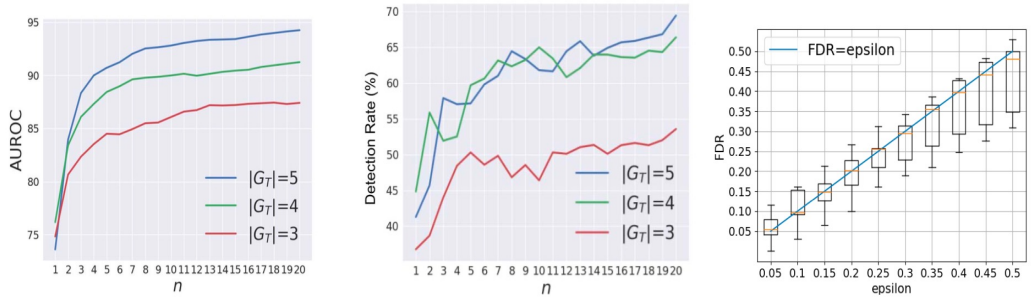


Figure 5: AUROC vs. n (left), TNR (with detection threshold at 95% TPR) vs. n (center) shows that the performance of CODiT increases with the increase in the number n of p-values used in the fisher-value for detection. False Alarm Rate (FDR) of CODiT is empirically bounded by ϵ on average (right). The yellow line in the box plot indicates the median.

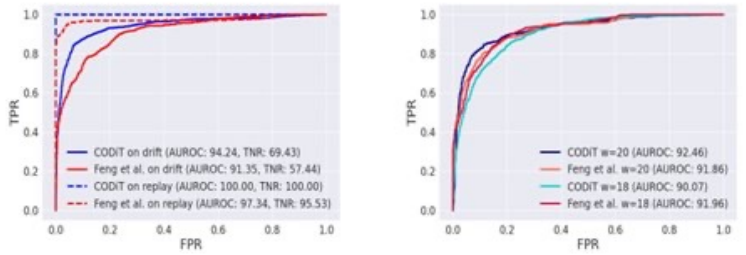


Figure 7: CODiT outperforms SOTA detector Feng et al. (2021) on temporal OODs in vision with the window length $w = 16$ (left). CODiT performs consistently well with different window lengths of $w = 18$, and 20 (right).

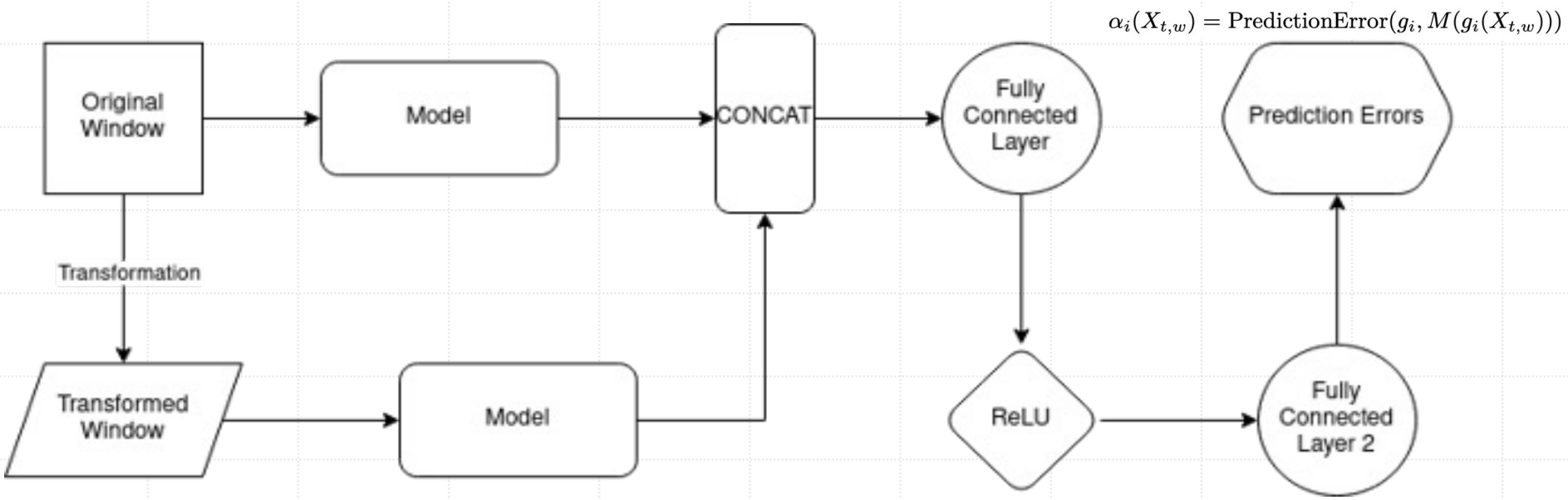
Table 2: Comparison of CODiT with Cai et al.'s (VAE), Ramakrishna et al.'s (β–VAE), and Feng et al.'s detectors on weather and night OODs from CARLA dataset. Best results are in bold.

OOD	AUROC (↑)				TNR (90% TPR) (↑)				Detection Delay (@95% TPR) (↓)			
	VAE	β–VAE	Feng's	Ours	VAE	β–VAE	Feng's	Ours	VAE	β–VAE	Feng's	Ours
Rainy	53.56	92.07	84.21	**99.71**	0	81.00	27.63	**98.57**	NA	**0.15**	5.33	**0.15**
Foggy	52.02	41.02	86.09	**99.66**	2.30	2.75	28.01	**98.05**	33.05	19.65	5.37	**0**
Snowy	53.23	**97.52**	95.91	96.67	0	**99.69**	78.20	86.47	NA	0.33	**0**	**0**
Night	50.86	95.57	75.07	**98.94**	1.78	71.90	0.40	**94.55**	72.80	4.07	85.4	**1.41**

Results (Contd..)

- The algorithm was evaluated on the following datasets –
  - Vision – Replay and Drift
  - Non-Vision – GAIT
- As seen in figure 4, the transformations consistently leave the In-Data loss unaffected and only affects the OOD loss, hence the equivariance learned from data is less likely to generalize to OOD windows
- The performance of the CoDIT algorithm improves consistently on increasing the number of p-values (n) as seen in figure 5 (left and center) above
- Increasing the number of transformations ($|G_T|$) also increases the overall performance
- We observe a bounded false alarm rate as intended (seen in figure 5, right)
- The variation of the performance is consistent with changes in the window length as seen in Fig 7
- The algorithm outperforms the SoTA methods on most datasets (Table 2)

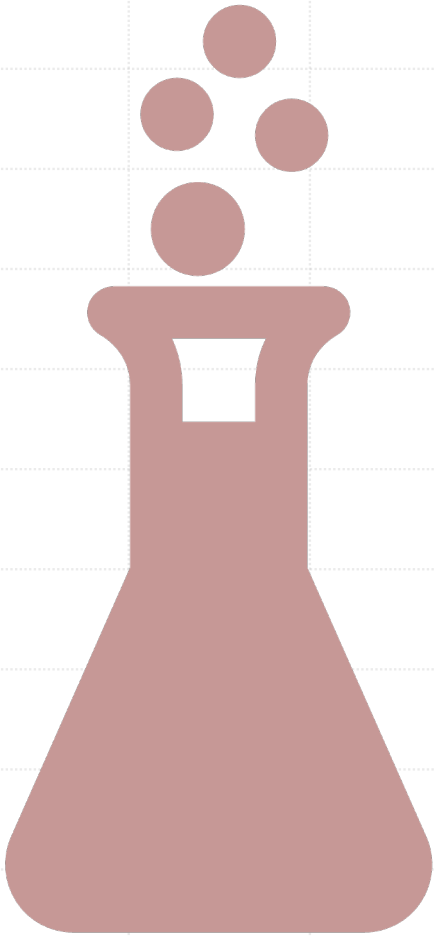
General Architecture



- Alphas (Prediction loss) are obtained from both Input Window and Calibration Set Windows, from which p-values are calculated.
- Multiple p-values are calculated for multiple transformations sampled independently from QGT (Transformation set). It's hypothesized that under one transformation, an OOD window might behave as the transformed iD window but the likelihood of this decreases with the number of transformations.
- These p-values (conformal predictions) are ensembled through Fisher's method.

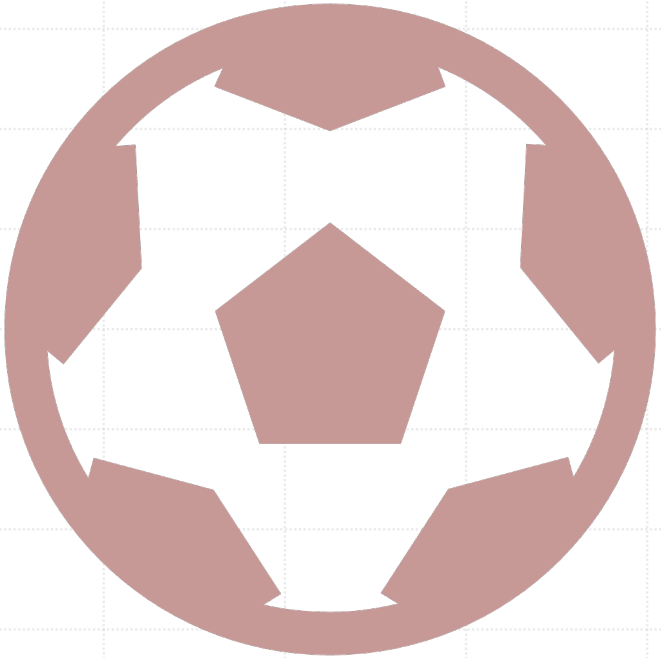
Verification and Experimentation

Code, Data and Trained Models are available at
https://github.com/kartik-iitk/ENCODiT



RoboCup Middle Size League

- International competition in which a team of five fully autonomous robots play with a FIFA-sized soccer ball
- Involves CPS spanning modules with LEC:
  - Swarm Robotics
  - Perception Systems
  - Localization Challenges
  - Decision Making
  - Path Planning



Challenges

Feedback Based Mechanisms might lead to erroneous outputs and unpredictable results

Methods to detect and handle hardware and sensor failures

Robust Ball trajectory prediction,

- Kicked ball can have an aerial or grounded trajectory in real life
- On ground, ball might show non-linear trajectory
- Estimation of ball trajectory is further used in decision making and game strategization

CODiT with the Ball Trajectory Prediction Model

Motivation

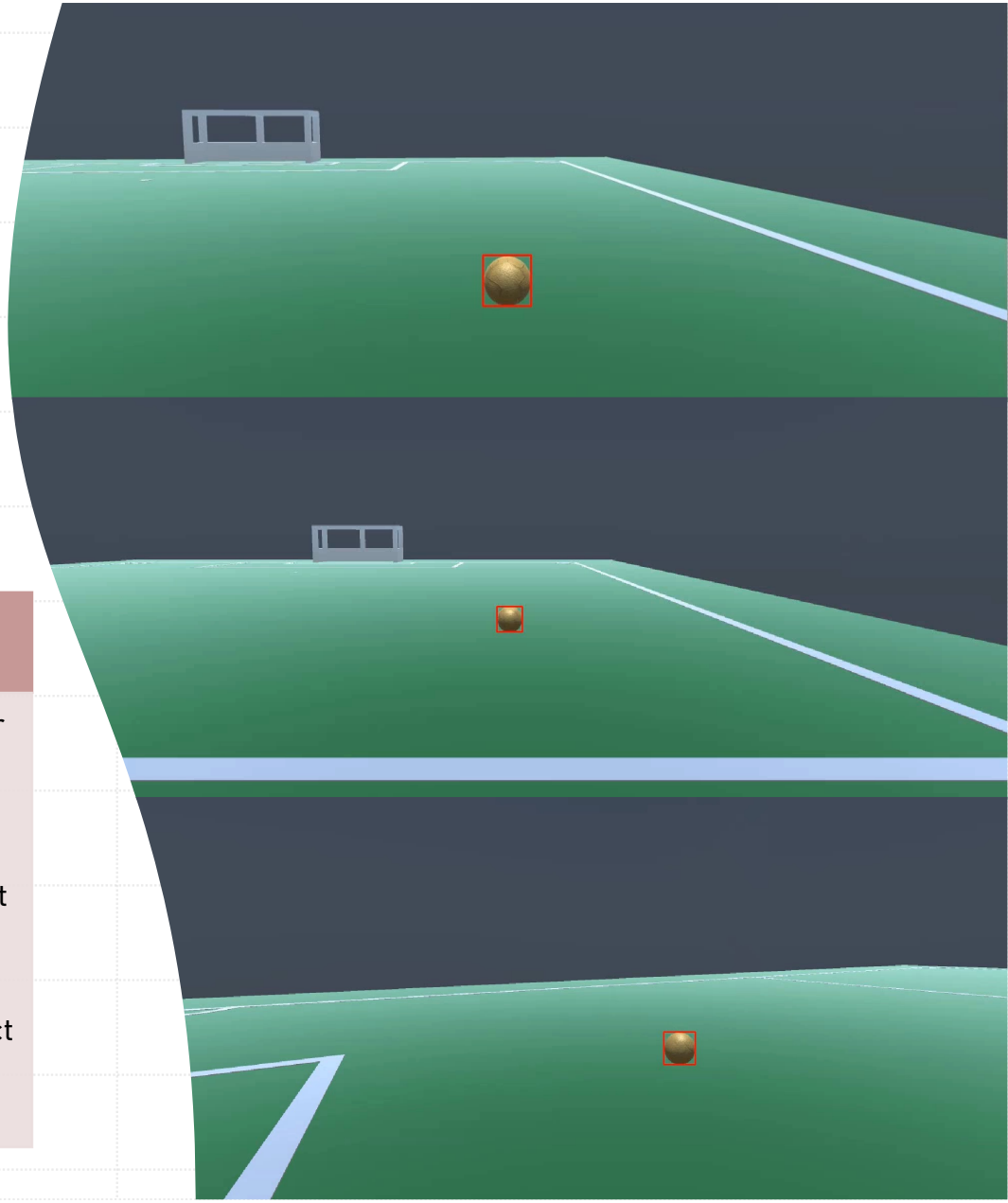
- OOD detection in Ball Trajectory Estimations can greatly benefit the **decision** module and prevent errors.
- CODiT works on **non-image data**, with a very low **computation overhead** helpful for edge devices.

Dataset Creation

- Made use of the **Unity Game Engine** for creating a simulation
- Generated **60 frame** video clip dataset with different kicking speeds and camera angles
- Test Dataset consisted of balls in a **curved** trajectory

Further Image Processing

- Dataset was further processed to **mask** and detect the soccer ball in video
- Bounding box was then used to extract the **x, y and ball area** (as a measure of depth) in time series data to detect OOD trajectories.



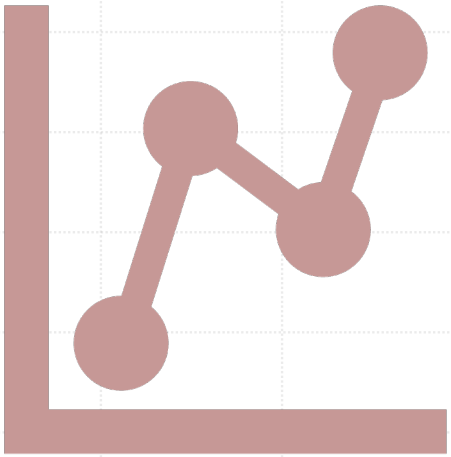
Dynamic Windows

Challenge

- Motivation: Decreasing the window size may improve the CPU Usage and Inference time of the overall pipeline of the decision module.
- It has not been implemented in the original code as it is difficult to incorporate it in the CNN architecture natively.
- This is because the window size has a direct impact on the layer parameters of the CNN.
- We countered this by training different models on different window sizes.
- We also formulated a unique GRU-CNN hybrid model to introduce a memory aspect as a step to do away with fixed window sizes altogether.

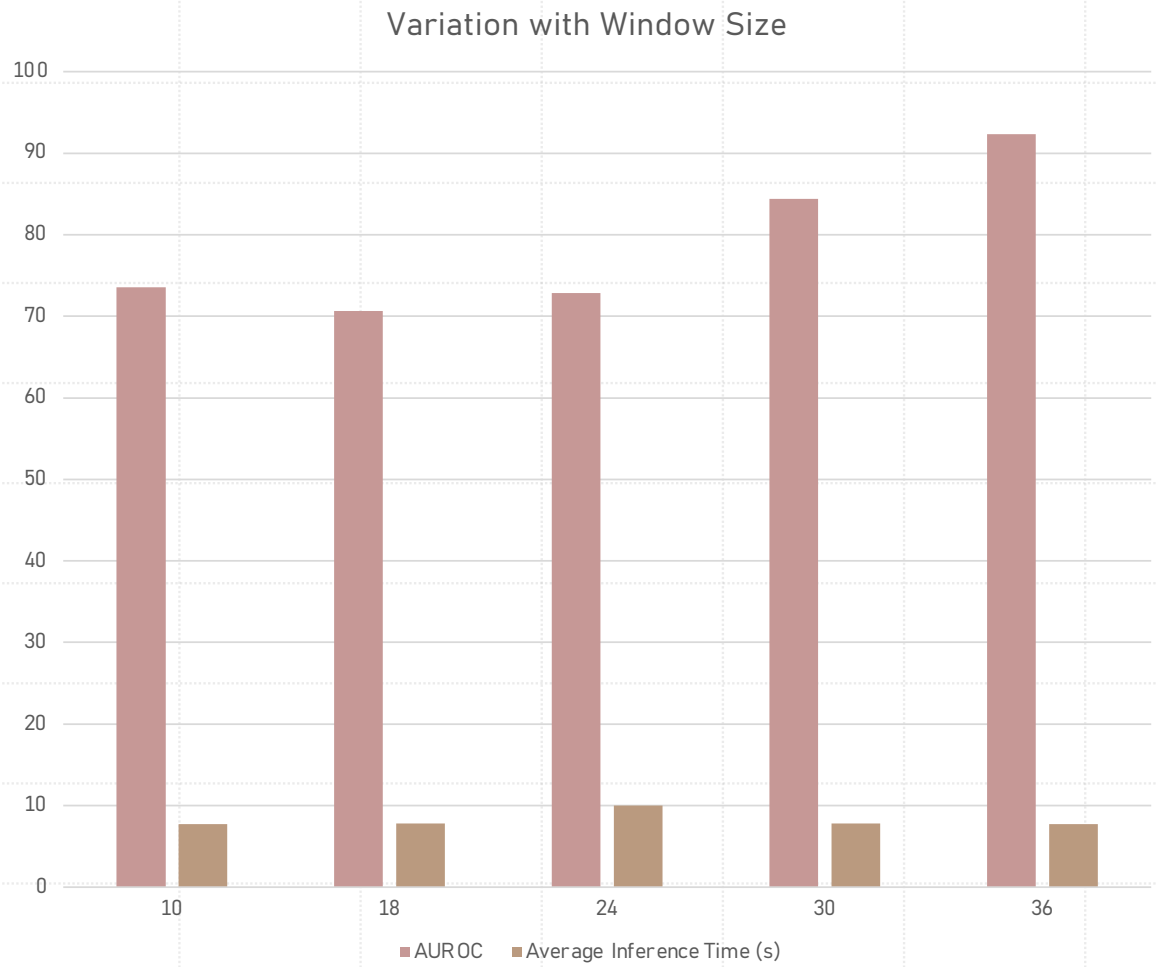
Experimentation

- We trained different LeNet models for different window lengths ranging from 10 to 36 size by the method illustrated in the paper.
- Performed Hyperparameter tuning for increasing evaluation metrics of the model.
- Since the code works by splitting the test data into train, valid and test_in segments using a random seed, we performed the dynamic window testing for a 10 different randomly chosen seeds and averaged out the inference time and the AUROC scores obtained.
- Further we plotted the variation of AUROC and Average Inference Time (s) versus the window length for different window lengths.



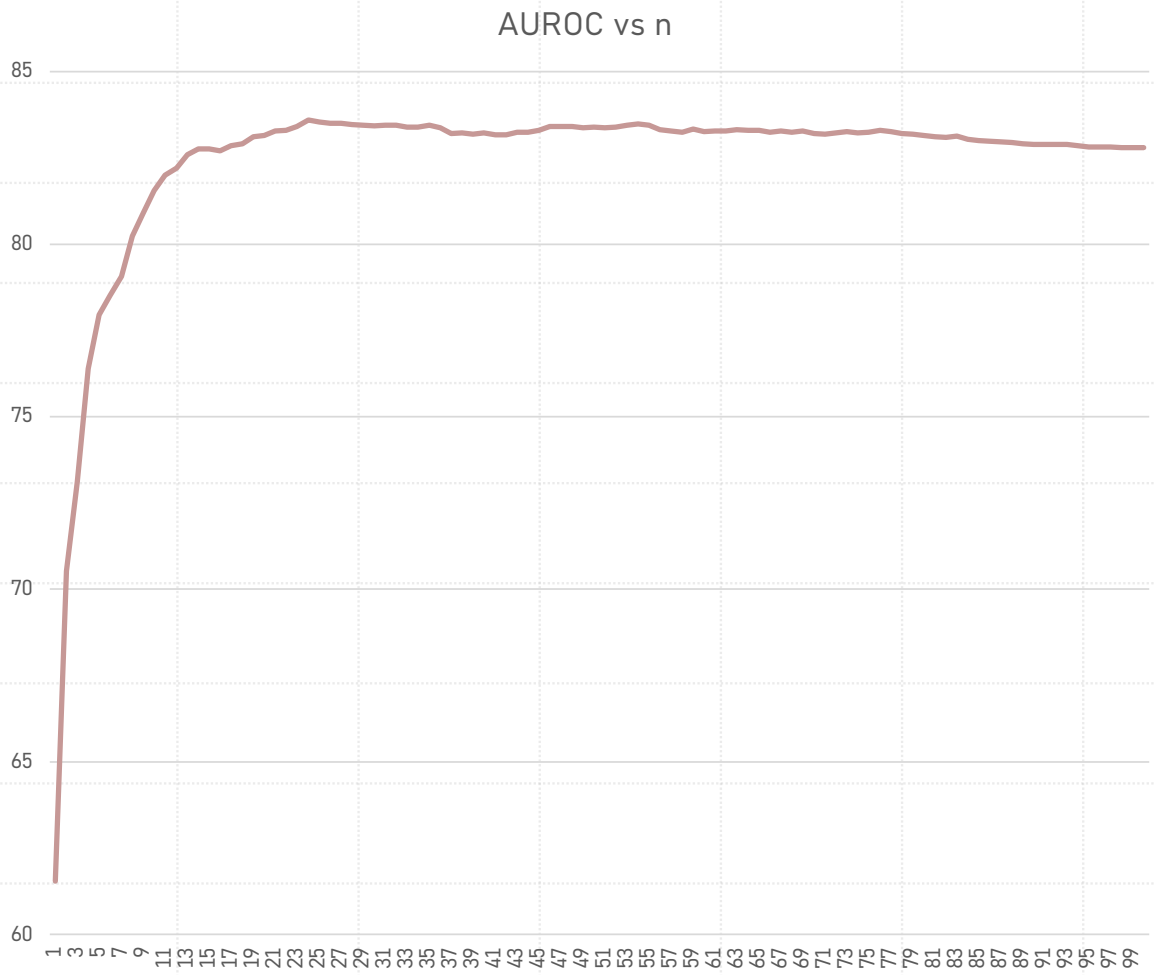
Variation with Window Size

- We observed that increasing window size usually always improves the output AUROC while the window size did not make any significant difference in inference time.
- The inference time and AUROC values showed consistent behavior on different machines



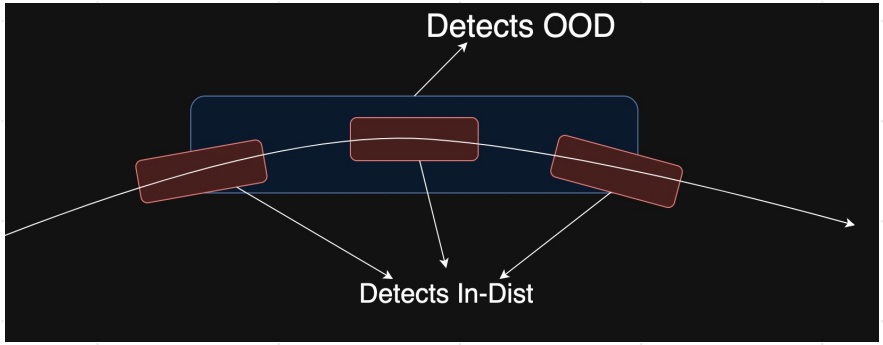
Variation with Number of Fisher Values for the custom LeNet5 Model with a fixed Window Size

- The original model proposed in the paper uses fisher-value of the input as the final OOD detection score
- A single p-value measures deviation from the iD GT -equivariance of the input with respect to one transformation $g \sim QGT$
- With multiple p-values, we test this deviation with respect to multiple transformations sampled independently from QGT
- We observe that detection performance of model increases as we increase the number n of p-values used in the final OOD detection score



RNN-based Approach

Extending to Sequential Models

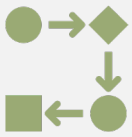


- Output of CPS systems are is continuous stream of data.
- Current architecture samples windows from the stream and compares it to in-order windows to decide if its OOD.
- This imposes a restriction of having a fixed window length, which might not give desired results in an unconstrained real environment, *for example*: In detection of OOD non-linear trajectories, a window with shorter length maybe detected as in-distribution whereas on increasing window length it is found to be OOD.

Extending to Sequential Models

Using sequential models following RNN architecture, we can make sampling window shorter, (5-20 instead of default 32-64) and utilise the hidden state, passing it as an input to the next step.

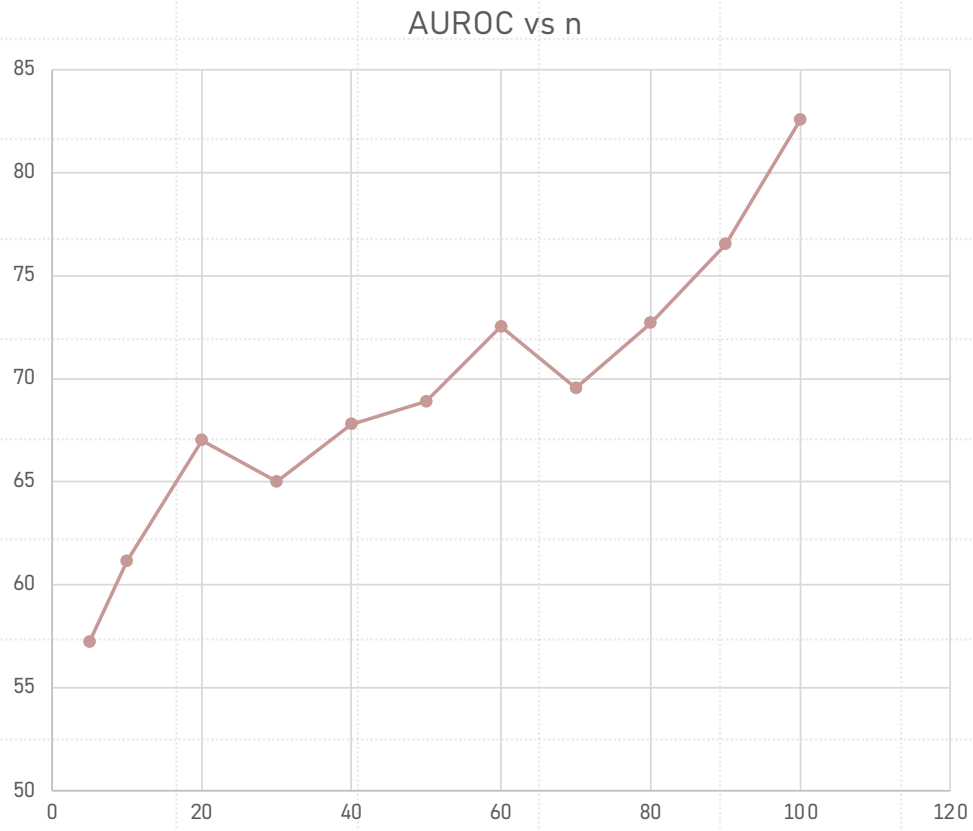
This helps us capture the temporal properties of data more effectively.



Same convolution-based transformation were being used, for which CNNs perform better, hence new transformations were needed. We tried [shuffling, periodic reversing]. There is potential in exploring better transformations capturing the in-distribution properties of data.

Experimentation

- We tried LSTMs, Hybrid models of GRU & CNNs and GRU with Self-Attention layers.
- While the GRU-CNN gave the highest validation accuracy during training, GRU with a self-attention layer in between gave the highest AUROC score (82.5).



Merits/ Demerits in the Approach

LSTM/GRU capture the temporality of the data continuously rather than quantized window sizes.

We propose using these in conjunction with attention layers to effectively capture temporal data and achieve dynamic window length behaviour, which should optimize performance and inference time significantly

While hybrid models of RNNs and attention layers are known to produce SOTA results on many time series task, it did not give the best results.

The primary reason was that the transformations proposed in the paper, viz., Erosion, Dilation, etc. are convolution-based, on which CNNs tend to perform better than RNNs

Another being that rather than labelling a time window as OOD, whole video is labelled as OOD, in which there are moments where ball is stationary for which model classifies it as in-distribution, but its label is OOD.

We tried exploring different possible transformations like [shuffling, reversing], but they gave sub-par results.

Future Work

Work on improving the LSTM's TNR and AUROC Score by exploring better transformations to aid in practical real-time applicability.

Use Ensemble-based models to improve post and pre-processing.

Consider usage of other temporal and non-temporal transformations to better learn the equivariance in our particular use case.

Expand and verify pipeline's use-case to different CPS.

Summary

Verification

- Ran original code for multiple random seeds and verified the accuracy of the algorithm in the paper.

RoboCup MSL

- Extended the algorithm to the novel use case for ball trajectory classification.
- Generated a custom dataset and implemented the algorithm.
- Modified the LeNet CNN model to get best performance of 92.314 AUROC for a window size of 36 on our custom dataset, which is close to the max accuracy.

Dynamic Windows

- Having a dynamic window could boost model performance and reduce inference time
- Trained different models for different window sizes
- No significant improvement in average inference time was obtained. Higher window size always gave a better AUROC and hence should be preferred.

RNN-Based Approach

- Formulated a unique GRU-CNN hybrid model to introduce a memory aspect as a step to do away with fixed window sizes altogether, which might not give desired results in an unconstrained real environment
- To leverage continuous data we implemented LSTMs, Hybrid models of GRU & CNNs and GRU with Self-Attention layers.
- GRU-CNN gave the highest validation accuracy during training, GRU with a self-attention layer in between gave the highest AUROC score (82.5)

Contribution

- **% Distribution:**
  - Kartik Anant Kulkarni (210493) – 25%
  - Rishi Agarwal (210849) – 25%
  - Emaad Ahmed (210369) – 25%
  - Dhruva Singh Sachan (210343) – 25%

Thank You