ANTI-MONEY LAUNDERING IN BITCOIN USING MACHINE LEARNING

*Amala Peter, *Karthika P V, *Maya Manohar K, *Parvathi Sankar, **Anuraj Mohan

*UG Scholar, **Assistant Professor, Department of Computer Science and Engineering

NSS College of Engineering, Palakkad

Anti-Money Laundering in Bitcoin

- Existing systems prove inefficient in tackling the issue of money laundering Bitcoin.
- The pseudonymity of Bitcoin is an advantage for criminals but the public availability of data is a key advantage for the investigators.

Objective

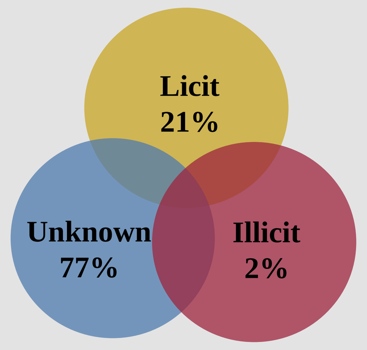
- Our work aims to exploit the publicly available data to develop useful insights that might help in curbing illegal activities.
- In this work, we experiment with various emerging methods that leverage graph information to model the problem and combine the potentialities of these methods to build a better performing system.
- We also aim to further improve our system using Knowledge Distillation (KD)

Problem Statement

To design an efficient system to classify the unknown transactions as licit or illicit in the Elliptic dataset to tackle the issue of money laundering in Bitcoin.

Elliptic Dataset

- 2,03,769 transactions/graph nodes and 2,34,355 edges representing the Bitcoin flow.
- 94 local features and 72 aggregate features.



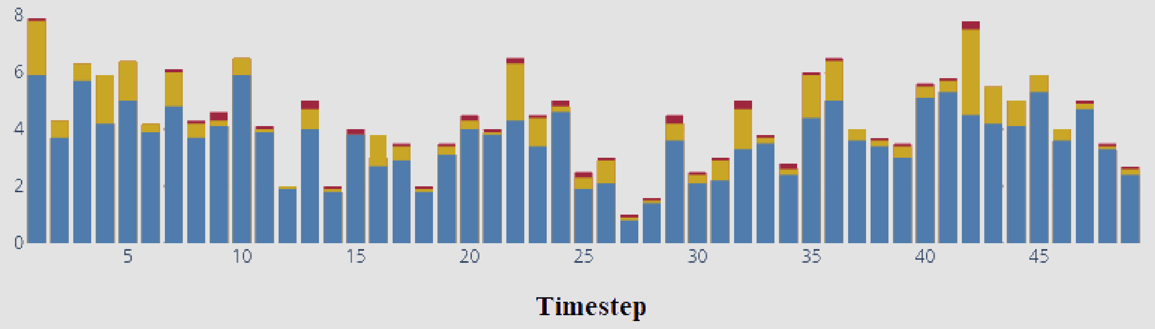


Fig. 1 Number of nodes vs. Timestep

Methodology - GCDF

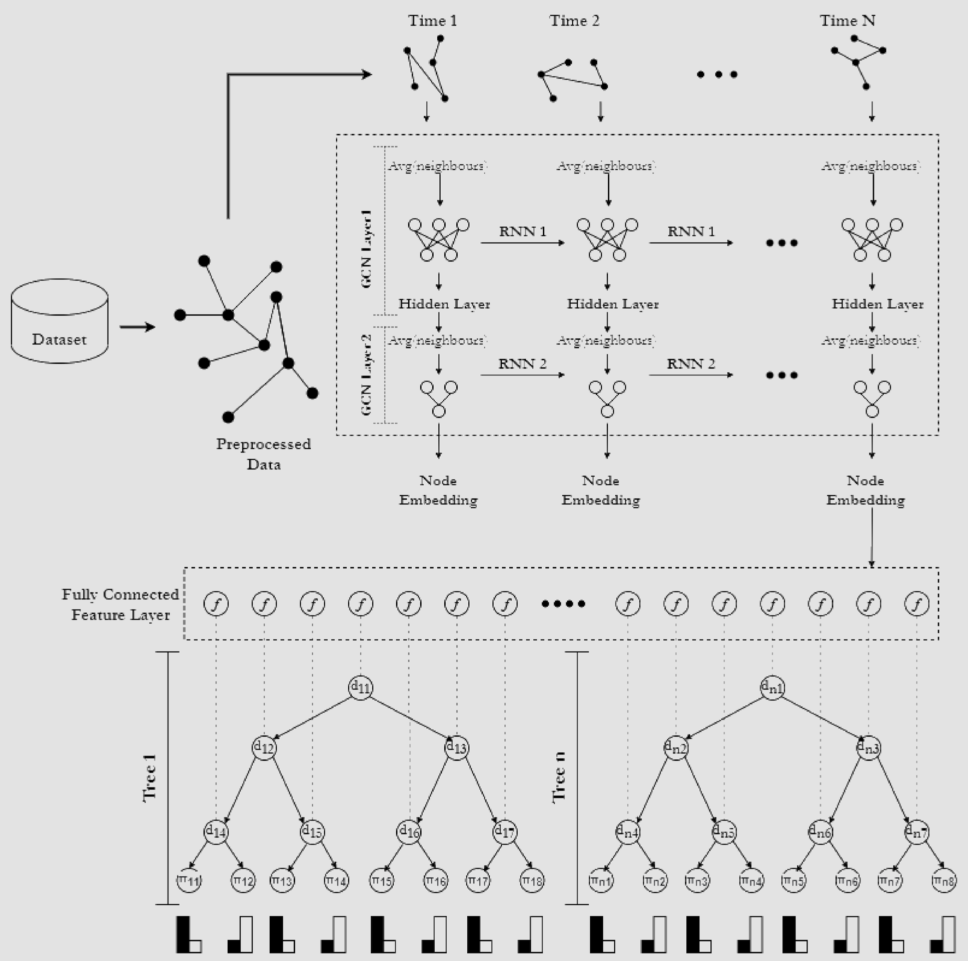


Fig 2. Proposed System - GCDF

Various steps involved in Graph Convolutional Decision Forest (GCDF) :

- Pre-process the dataset
- Feed each timestep to EvolveGCN module
- Feed the node embeddings obtained from EvolveGCN to Deep Neural Decision Forest (DNDF) Module
- Obtain the final prediction

Fine Tuning using KD

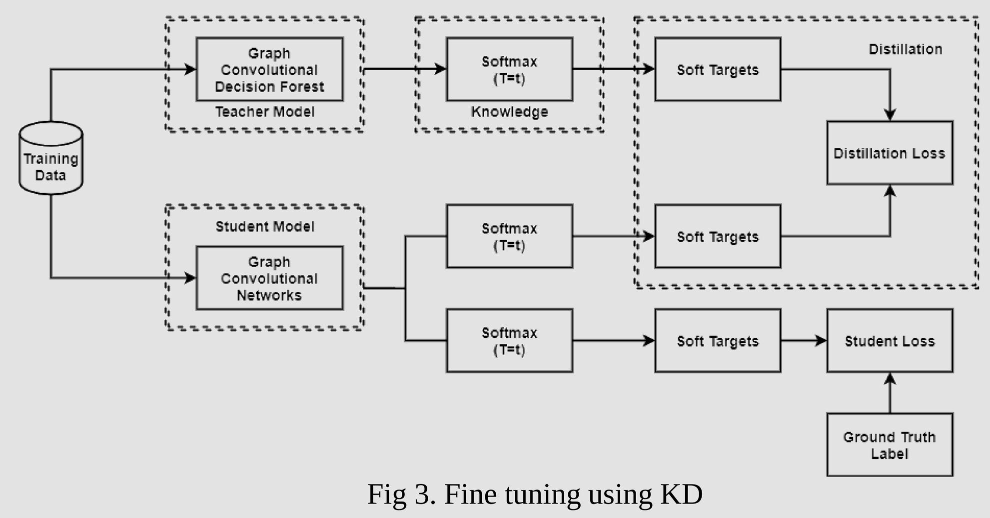


Fig 3. Fine tuning using KD

Various steps involved in fine tuning :

- Train GCDF as the teacher model and obtain the distillation loss
- Train GCN as the student model using the distillation loss
- Obtain final predictions from the student model

Tools

- Python
- NumPy
- Sklearn
- PyTorch

Evaluation Measures

- Precision
- Recall
- F1-Score
- Micro Average F1-Score

Results

Perfomance Comparison	Illicit			Micro Avg F1
	Precision	Recall	F1	
Logistic Regression (AF + NE)	0.457	0.651	0.537	0.9297
Random Forest (AF + NE)	0.984	0.647	0.781	0.9772
MLP (AF + NE)	0.784	0.542	0.641	0.9619
Graph Convolutional Network	0.8674	0.4774	0.6158	0.9613
GraphSAGE	0.8534	0.8385	0.8939	0.8278
EvolveGCN	0.998	0.8663	0.9249	0.8663
GCDF	0.9953	0.8663	0.9251	0.8663

Table. 1 GCDF vs. Other Methods; AF – All Features, NE – Node Embeddings

Methods	Illicit			Micro-avg F1
	F1 score	Precision	Recall	
GCDF (Without KD)	0.9251	0.9953	0.8663	0.8663
GCDF (With KD) - T	0.9251	0.9953	0.8663	0.8663
GCDF (With KD) - S	0.9525	0.9936	0.9166	0.9191

Table. 2 Effect of KD on GCDF

Methods	Teacher				Student			
	F1 Score	Precision	Recall	Micro-avg F1	F1 Score	Precision	Recall	Micro-avg F1
GCN	0.444	0.305	0.406	0.9946	0.8175	0.7828	0.8751	0.708
EvolveGCN	0.9251	0.9931	0.8663	0.8663	0.9252	0.9999	0.8666	0.8666
GCDF	0.9251	0.9953	0.8663	0.8663	0.9525	0.9936	0.9166	0.9191

Table. 3 Other Methods in KD

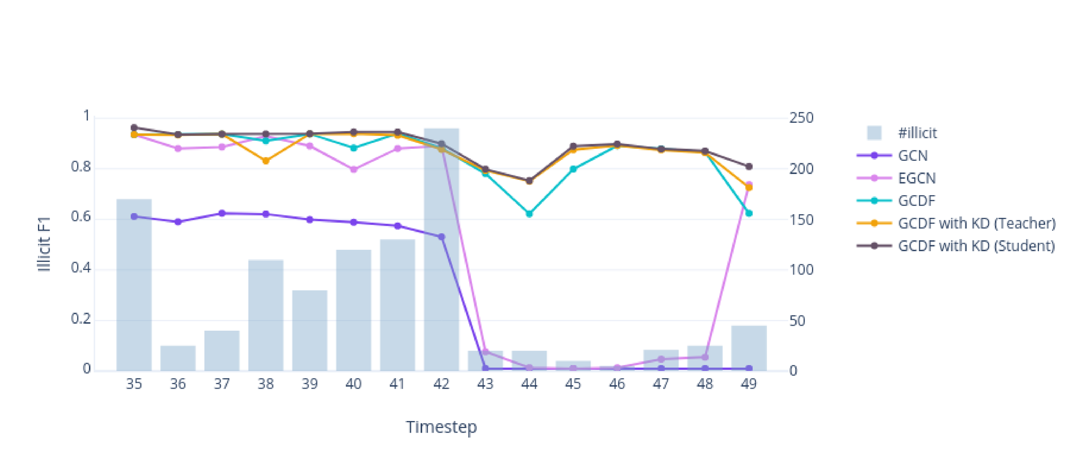
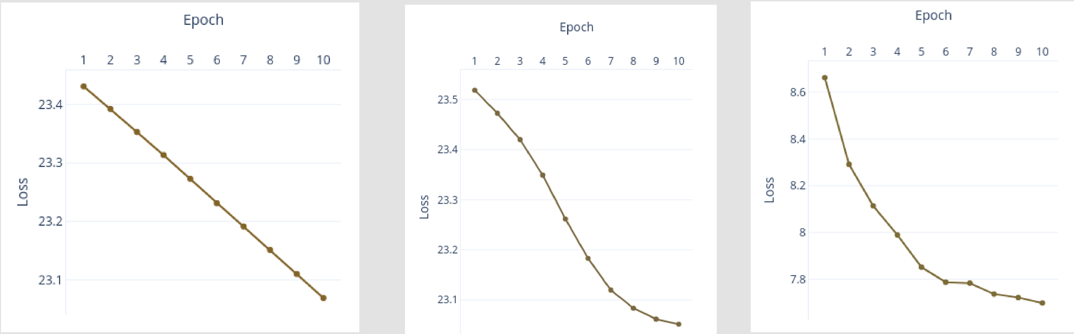


Fig 4. Illicit F1 results over timespan



(a)GCDF (b)GCDF with KD (T) (c)GCDF with KD (S)

Fig 5. Loss vs. Epoch ; T-Teacher S-Student

Fig 6. Illicit F1 vs. Tree size in DNDF

Inferences

- GCDF performs best with a 70:30 temporal split of training and test data respectively.
- 80 trees of depth 8 each in DNDF was able to give satisfactory results both in terms of performance and execution time.
- Loss incurred while training student convincingly reduced with the introduction of KD
- There was an observable performance boost in the student model as compared to the teacher model.

Conclusion

- Out of the benchmark methods, Random Forest gives the best result. But this does not incorporate any graph information
- Our proposed system is implemented as a combination of Random Forests and graph information.
- With the notion of appending dynamicity to the model, the dynamic method of EvolveGCN was used by replacing GCN which is static.
- Additionally the application of KD gave finer results

Future Works

- The decision trees may be inaccurate comparatively and their instability may lead to large structural changes.
- Elliptic dataset has the main limitation of having a new node set for each new graph snapshot; this needs to be addressed while considering a dynamic setting.
- Our future work will be with the intention to explore any other publicly available dataset and attempt novel dynamic techniques on those.

References

[1] Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T. and Leiserson, C.E., 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics.

[2] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. and Leiserson, C., 2020, April. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In Proceedings of the AAAI Conference on Artificial Intelligence

[3] Kontschieder, P., Fiterau, M., Criminisi, A. and Bulo, S.R., 2015. Deep neural decision forests. In Proceedings of the IEEE international conference on computer vision (pp. 1467-1475).

[4] Yang, Y., Qiu, J., Song, M., Tao, D. and Wang, X., 2020. Distilling knowledge from graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7074-7083).