北京郵電大學
Beijing University of Posts and Telecommunications

UIC
UNIVERSITY OF ILLINOIS AT CHICAGO

Recent Developments of Deep Heterogeneous Information Network Analysis --Part IV： Applications

Chuan Shi

shichuan@bupt.edu.cn

Beijing University of Posts and Telecommunications

Philip S. Yu

psyu@uic.edu

University of Illinois at Chicago

Outline

- Metapath based data mining
- Heterogeneous information network embedding
- ✓ **Applications**
  - HinDroid (KDD2017), HACUD (AAAI2019), MEIRec (KDD2019)
- Conclusion and future work

HinDroid

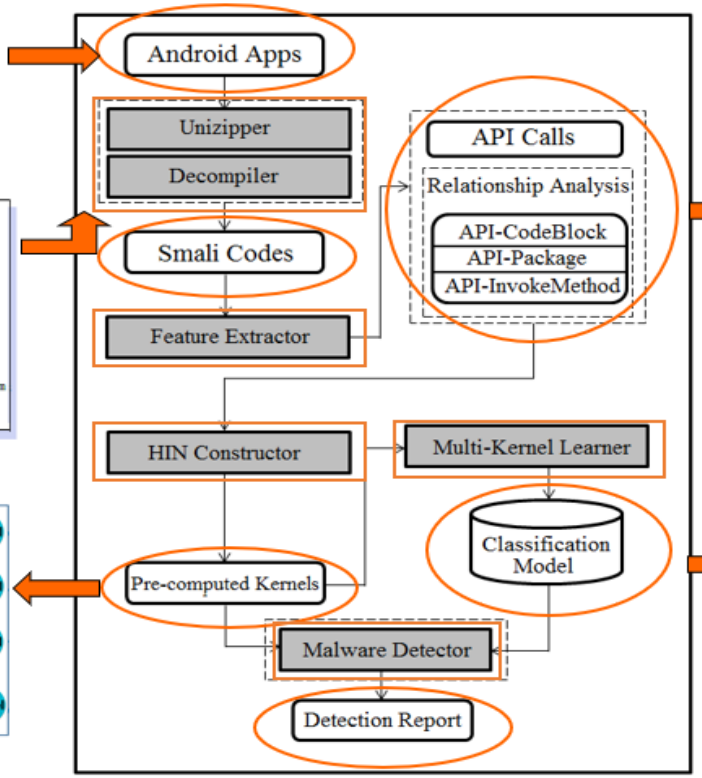
- HinDroid: a more resilient system that helps protect smart phone users against Android malware attacks and novel threats.

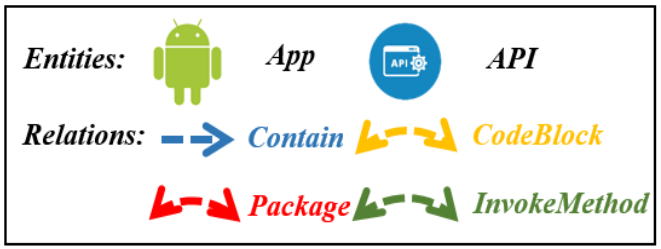
Android app is compiled and packaged in a single archive file (.apk) that includes the app **code** (.dex file), resources, assets, and manifest file.

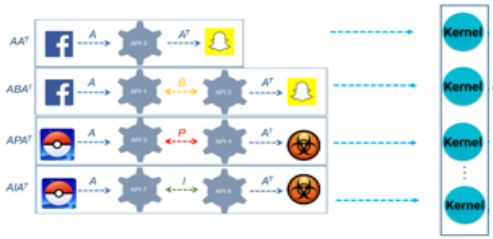
Listing 1: An example of smali code

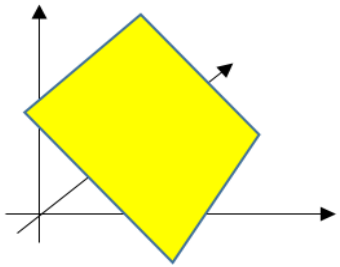
```
1  .method protected
2  loadLibs(Landroid/content/Context;)V
3  .locals 4
4  :try_start_0
5  new-instance v0, Ljava/io/BufferedReader;
6  new-instance v1, Ljava/io/InputStreamReader;
7  invoke-static {}, Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;
8  move-result-object v2
9  const-string v3, "getprop ro.product.cpu.abi"
10 invoke-virtual {v2, v3}, Ljava/lang/Runtime;->exec(Ljava/lang/String;)
       Ljava/lang/Process;
11 move-result-object v2
12 invoke-virtual {v2}, Ljava/lang/Process;->getInputStream()Ljava/io/InputStream
13 ......
14 .end method
```

HinDroid System Architecture







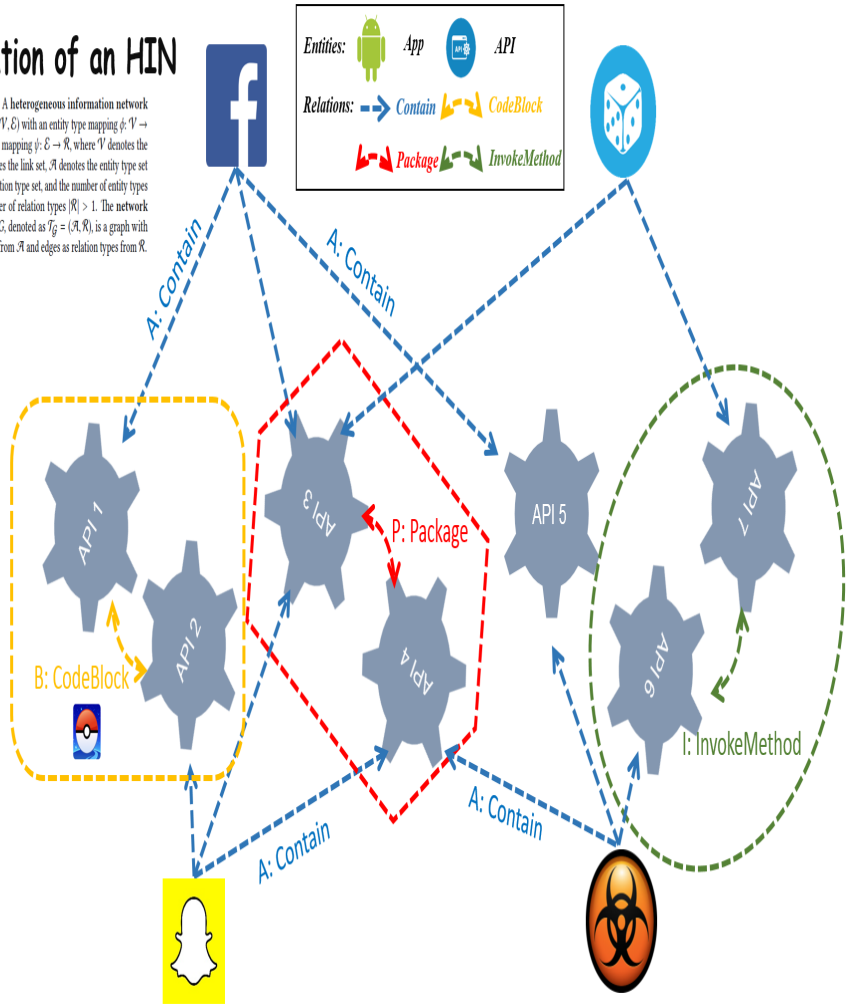


Shifu Hou, Yanfang Ye, Yangqiu Song, Melih Abdulhayoglu. HinDroid: An Intelligent Android Malware Detection System Based on Structured Heterogeneous Information Network. KDD 2017 (Best Application Paper).

HIN and Metapaths

Illustration of an HIN

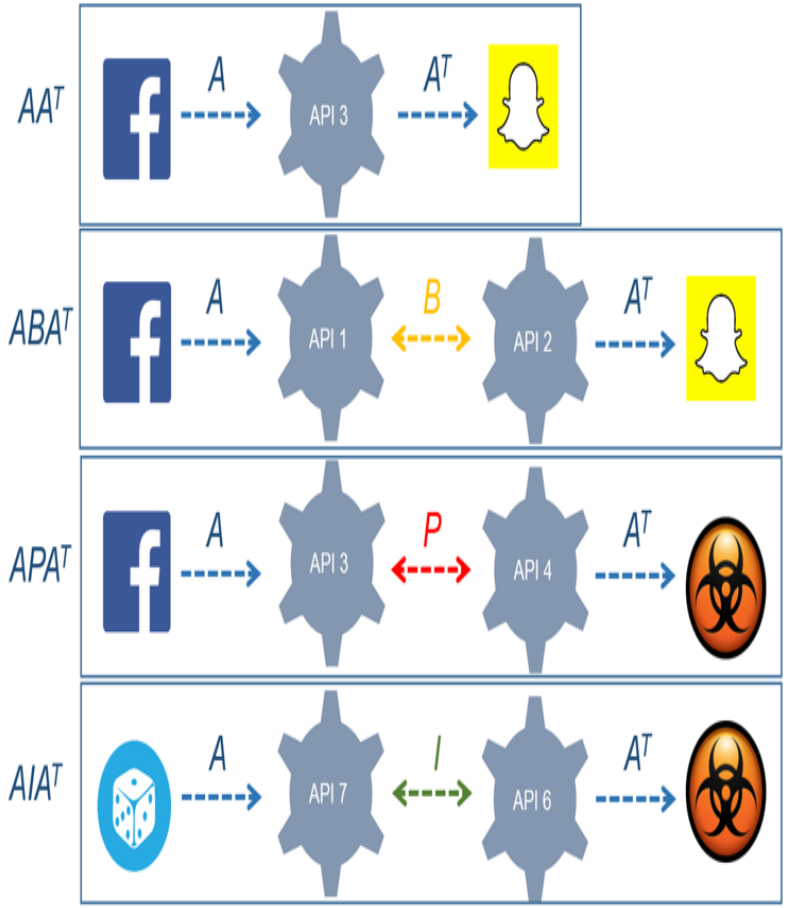
Definition 3.1. [18] A **heterogeneous information network** (HIN) is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an entity type mapping $\phi: \mathcal{V} \to \mathcal{A}$ and a relation type mapping $\psi: \mathcal{E} \to \mathcal{R}$, where $\mathcal{V}$ denotes the entity set and $\mathcal{E}$ denotes the link set, $\mathcal{A}$ denotes the entity type set and $\mathcal{R}$ denotes the relation type set, and the number of entity types $|\mathcal{A}| > 1$ or the number of relation types $|\mathcal{R}| > 1$. The **network schema** for network G, denoted as $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$, is a graph with nodes as entity types from $\mathcal{A}$ and edges as relation types from $\mathcal{R}$.



Meta-path Generation

Definition 3.2. [19] A **meta-path** $\mathcal{P}$ is a path defined on the graph of network schema $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$, and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{L+1}$, which defines a composite relation $R = R_1 \cdot R_2 \cdot \dots \cdot R_l$ between types A_1 and A_{L+1}, where $\cdot$ denotes relation composition operator, and L is the length of $\mathcal{P}$.

Definition 3.3. [19] Given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and its network schema $\mathcal{T}_G$, a **commuting matrix** $M_\mathcal{P}$ for a meta-path $\mathcal{P} = (A_1 - A_2 - \dots - A_{L+1})$ is defined as $M_\mathcal{P} = G_{A_1A_2}G_{A_2A_3}\dots G_{A_LA_{L+1}}$, where $G_{A_iA_j}$ is the adjacency matrix between types A_i and A_j. $M_\mathcal{P}(i, j)$ represents the number of path instances between entities $x_i \in A_1$ and $y_j \in A_{L+1}$ under the meta-path $\mathcal{P}$.



Experimental Results and Analysis

E1: Detection Performance Evaluation of the Proposed Method

Table 3: Detection performance evaluation

PID	Method	F1	β	ACC	TP	FP	TN	FN
1	AA^T	0.9529	0.1069	94.40%	283	19	189	19
2	ABA^T	0.9581	0.0900	95.00%	286	9	189	16
3	APA^T	0.9495	0.0858	94.20%	273	0	198	29
4	AIA^T	0.9183	0.0623	90.40%	270	16	182	32
5	$ABPB^TA^T$	0.9479	0.0670	94.00%	273	1	197	29
6	$APBP^TA^T$	0.9502	0.0565	94.20%	277	4	194	25
7	$ABIB^TA^T$	0.8683	0.0639	84.60%	254	29	169	48
8	$AIBI^TA^T$	0.8722	0.0639	85.00%	256	29	169	46
9	$APIP^TA^T$	0.8373	0.0445	81.20%	242	34	164	60
10	$AIPI^TA^T$	0.8761	0.0572	86.60%	237	2	196	65
11	$ABPIP^TB^TA^T$	0.9184	0.0616	90.80%	259	3	195	43
12	$APBIB^TP^TA^T$	0.8597	0.0617	84.60%	236	11	187	66
13	$ABIPI^TB^TA^T$	0.9284	0.0426	91.80%	266	5	193	36
14	$AIBPB^TI^TA^T$	0.8237	0.0426	82.60%	218	3	195	84
15	$AIPBP^TI^TA^T$	0.8597	0.0469	81.60%	215	5	193	87
16	$APIBI^TP^TA^T$	0.8597	0.0458	84.60%	236	11	187	66
17	Combined-kernel (5)	0.9214	—	91.20%	258	0	198	44
18	Combined-kernel (16)	0.9740	—	96.80%	300	14	184	2
19	Multi-kernel (5)	0.9834	—	98.00%	297	5	193	5
20	Multi-kernel (16)	**0.9884**	—	**98.60%**	299	4	194	3

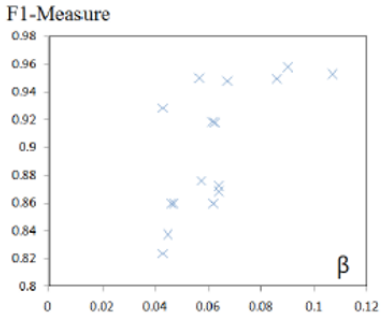


Figure 3: β_k and F1 correlation.

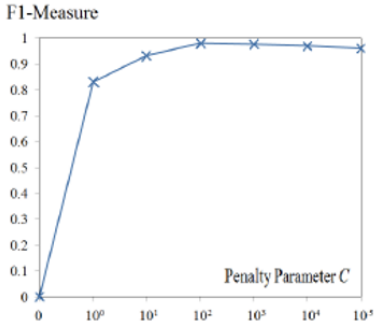


Figure 4: Parameter sensitivity evaluation.

Remark: In Figure 3, β_k is the parameter learned by multi-kernel learning, shown in Eq. (1). F1 is the actual performance of SVM using each meta-path as kernel.

Combined-kernel: We rank each meta-path using its Laplacian score. The order of the ranking is: *PID12 -> PID16 -> **PID6** -> **PID3** -> **PID5** -> PID11 -> PID9 -> PID2 -> PID8 -> PID7 -> PID13 -> PID14 -> PID15 -> PID10 -> PID4 -> PID1.*

Experimental Results and Analysis

E2: Comparisons of HinDroid and other Alternative Detection Methods

Table 4: Comparisons between HinDroid and alternative detection methods. "Original" means all the algorithms use original app features (i.e., API calls) as input. "Augmented" means that, we simply put all HIN-related entities and relations as features for different algorithms to learn.

Original	F1	AUC	ACC	TP	FP	TN	FN
ANN-1	0.9173	0.9023	90.20%	272	19	179	30
NB-1	0.8514	0.8511	83.60%	235	15	183	67
DT-1	0.9202	0.9005	90.40%	277	23	175	25
SVM-1	0.9529	0.9458	94.40%	283	9	189	19
Augmented	**F1**	**AUC**	**ACC**	**TP**	**FP**	**TN**	**FN**
ANN-2	0.9409	0.9316	93.00%	279	12	186	23
NB-2	0.9025	0.8891	88.60%	264	19	179	38
DT-2	0.9539	0.9397	94.40%	290	16	182	12
SVM-2	0.9590	0.9537	95.20%	281	7	191	17
HinDroid	**0.9884**	**0.9849**	**98.60%**	299	4	194	3

For ANN, we use 3 hidden layers (500 neurons in each hidden layer) and train the network using back propagation. The learning rate is set to 0.3 and the momentum is set as 0.5. For SVM, we use LibSVM in our experiment and the penalty is empirically set to be 1,000.

Table 5: Comparisons with other mobile security products

Family	Sample #	Norton	Lookout	CM	***HinDroid***
Lotoor	78	75	74	76	78
RevMob	52	46	50	48	52
Malapp	33	29	32	30	33
Fakebank	31	29	30	29	30
Generisk	29	29	29	29	29
GhostPush	19	15	16	18	18
Fakegupdt	16	15	14	14	16
Danpay	21	19	20	20	21
HideIcon	12	11	9	8	12
Idownloader	11	10	9	9	10
Total	302	278	283	281	**299**
DetectionRate	–	92.05%	93.71%	93.05%	**99.01%**

For the comparisons, we use all the latest versions of the mobile security products (i.e., Clean Master (CM): 2.08, Lookout: 10.9-7f33b3e, and Norton: 3.17.0.3205)

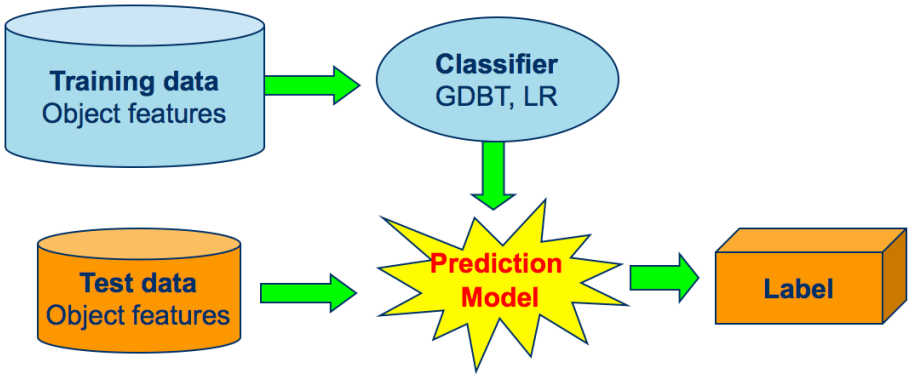
Cash-out User Detection

- **Credit Payment Services**
  - Credit card services in commercial banks
  - Credit payments in Internet financial institutions
- **Cash-out Fraud:** pursue cash gains with illegal means
  - E.g., buying pre-paid cards then reselling them.
- **Cash-out User Detection**
  - Predict whether a user will do cash-out transactions

Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, Yuan Qi. Cash-out User Detection based on Attributed Heterogeneous Information Network with a Hierarchical Attention Mechanism. AAAI 2019.

Background

- **Conventional solutions**



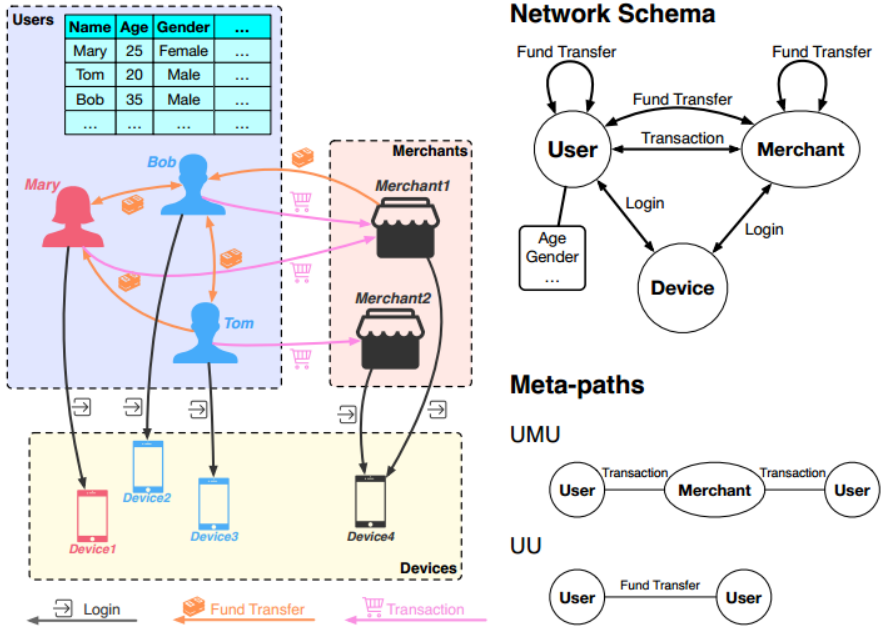
- Key: feature extraction
- Shortcoming: seldom fully exploit the interaction relations

- **Solution: integrate more auxiliary information, e.g.,**
  - The fund transfer relation among users and merchants
  - The login relation between users and devices
  - Abundant attribute information

Attributed heterogeneous information network is a promising way to integrate auxiliary data.

Background

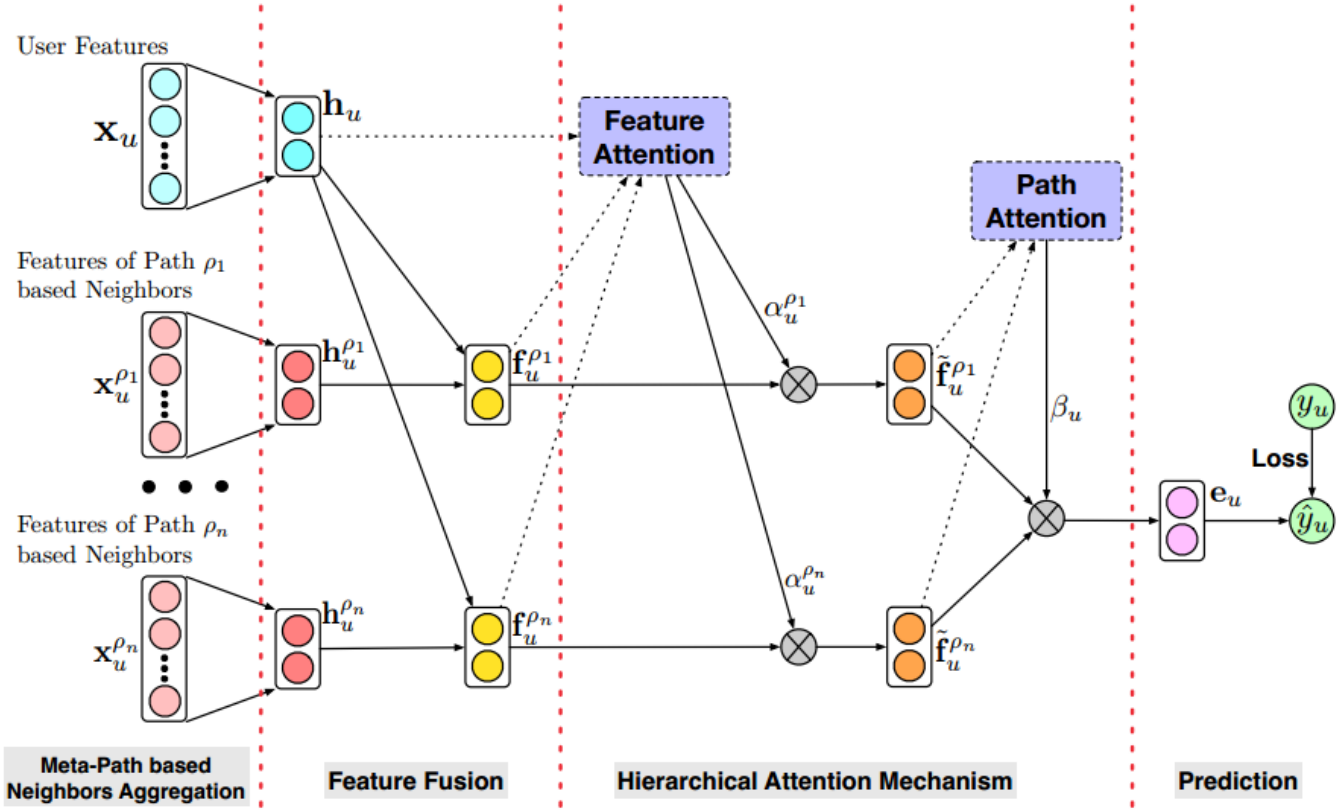
- **Attributed Heterogeneous Information Network (AHIN)**
  - Include multiple types of nodes or links and rich attribute information
  - Flexibly characterize heterogeneous data
  - Contain rich semantics
- **Meta-path**
  - A relation sequence connecting two objects in HIN
  - Extract structural features
  - Embody path semantics



(a) Scenario of credit payment service (b) Network schema and meta-path examples

Framework of HACUD

Hierarchical Attention mechanism based Cash-out User Detection model (HACUD)



Experiment

Datasets

- **Ten Days Dataset**

  1.88 million users 2018/03/21-2018/03/31

- **One Month Dataset**

  5.16 million users 2018/03/01-2018/03/31

Metrics

$$AUC = \frac{\sum_{u \in \mathcal{U}^+} rank_u - \frac{|\mathcal{U}^+| \times (|\mathcal{U}^+|+1)}{2}}{|\mathcal{U}^+| \times |\mathcal{U}^-|}.$$

Methods to Compare

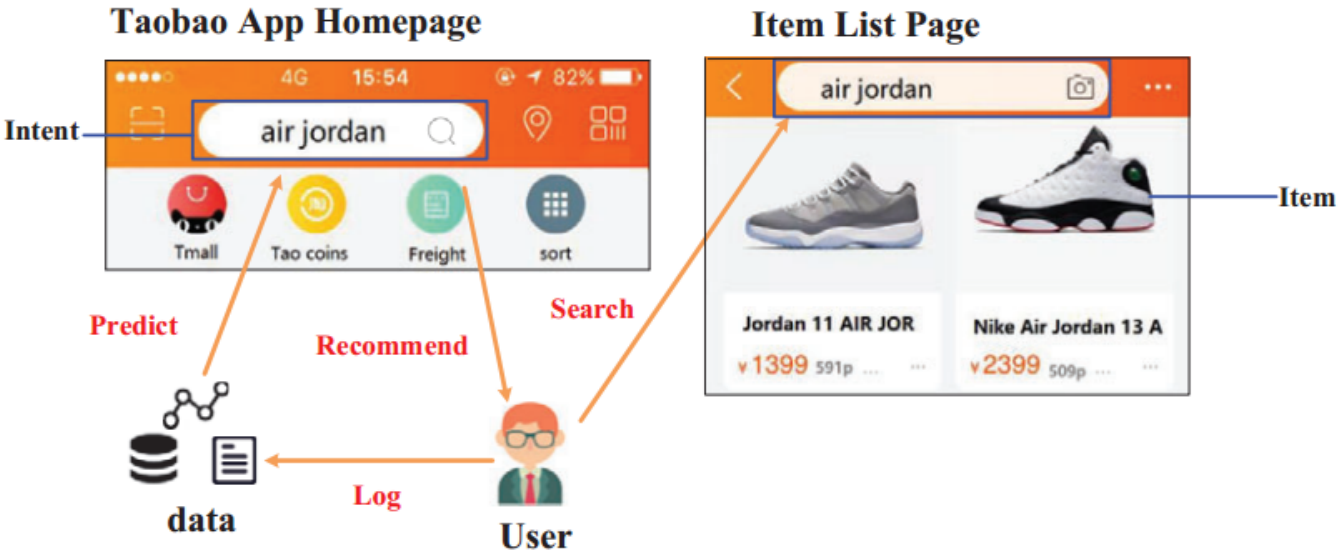
- **Attribute only or Structure only**
  - GBDT
  - Node2vec
  - Metapath2vec
- **Structure + Attribute**
  - Node2vec + Feature
  - Metapath2vec + Feature
- **Structure + Attribute + Label**
  - Structure2vec
  - $GBDT_{Struct}$

Effectiveness Experiments

Algorithm	AUC							
	Ten Days Dataset				One Month Dataset			
	$d = 16$	$d = 32$	$d = 64$	$d = 128$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
Node2vec	0.5893	0.5913	0.5926	0.5930	0.5980	0.6963	0.6009	0.6021
Metapath2vec	0.5914	0.5903	0.5917	0.5920	0.6005	0.5976	0.5995	0.5983
Node2vec + Feature	0.6455	0.6464	0.6510	0.6447	0.6541	0.6561	0.6607	0.6518
Metapath2vec + Feature	0.6456	0.6429	0.6469	0.6485	0.4850	0.6552	0.6523	0.6545
Structure2vec	0.6537	0.6556	0.6598	0.6545	0.6641	0.6632	0.6657	0.6678
GBDT	0.6389	0.6389	0.6389	0.6389	0.6467	0.6467	0.6467	0.6467
GBDT_{Struct}	0.6948	0.6948	0.6948	0.6948	0.6968	0.6968	0.6968	0.6968
HACUD	**0.7066**	**0.7115**	**0.7056**	**0.7049**	**0.7132**	**0.7160**	**0.7109**	**0.7154**

Background

- **Intent recommendation**
  - A new recommendation service in many mobile e-commerce Apps.
  - Automatically recommend a personalized intent for a user according to his/her historical behaviors without query input.



Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, Yongliang Li, Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation. KDD 2019.

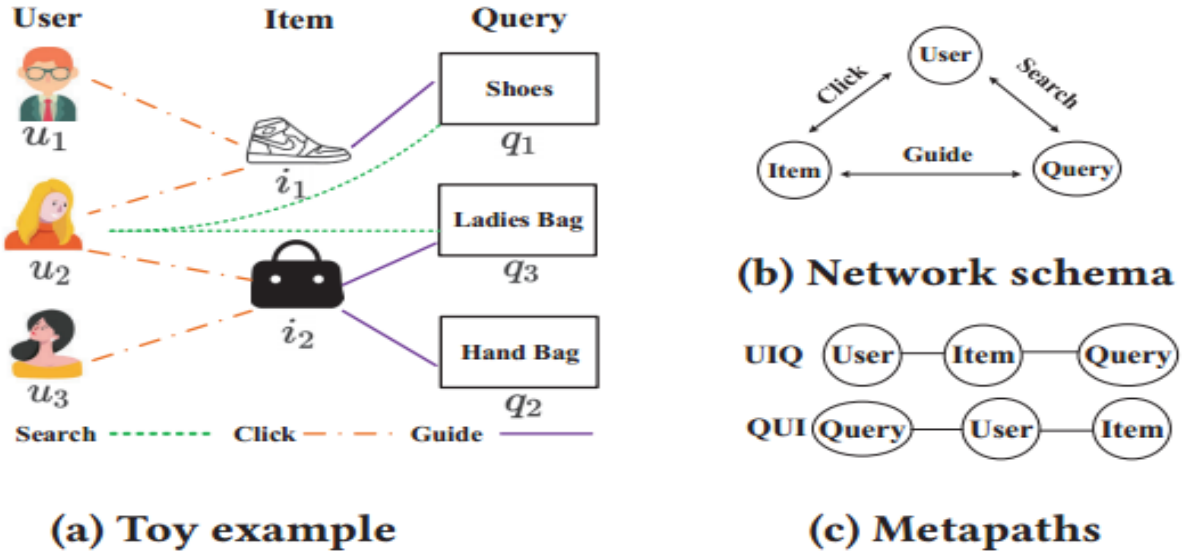
Background

Existing methods used in industry

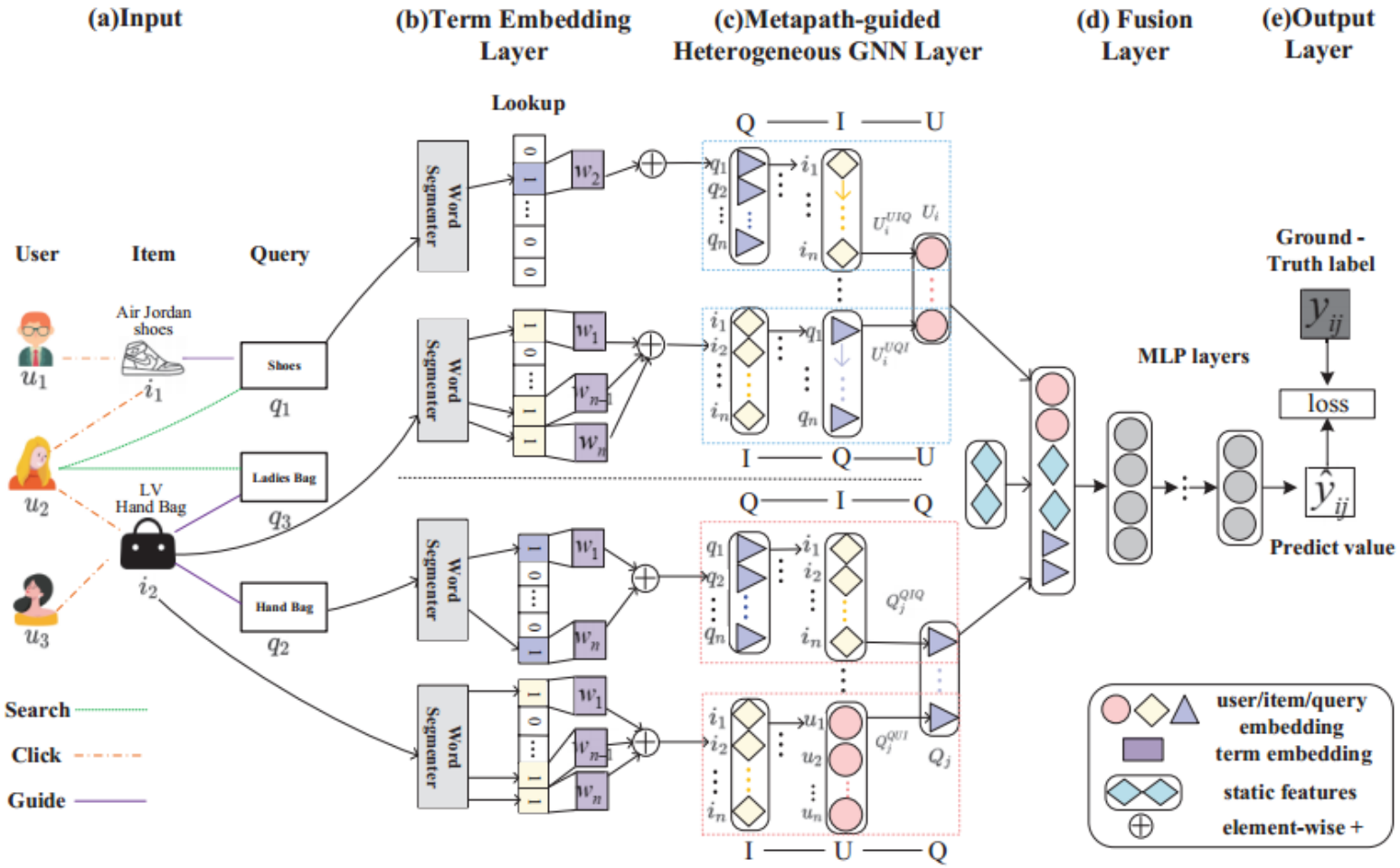
- **Classification method**
  - heavily rely on domain knowledge and need laboring feature engineering
  - fail to take full advantage of rich interaction information
- **Item recommendation method**
  - Only consider binary interactions between users and items
  - Only consider atomic and static items, intent always dynamic change.

MEIRec

- **Model intent recommendation with a HIN**
  - Flexibly exploit rich interaction
- **Heterogeneous Graph Neural Network**
  - Learn structural feature representations of users and queries
  - A uniform term embedding mechanism to handle large-scale and dynamic data

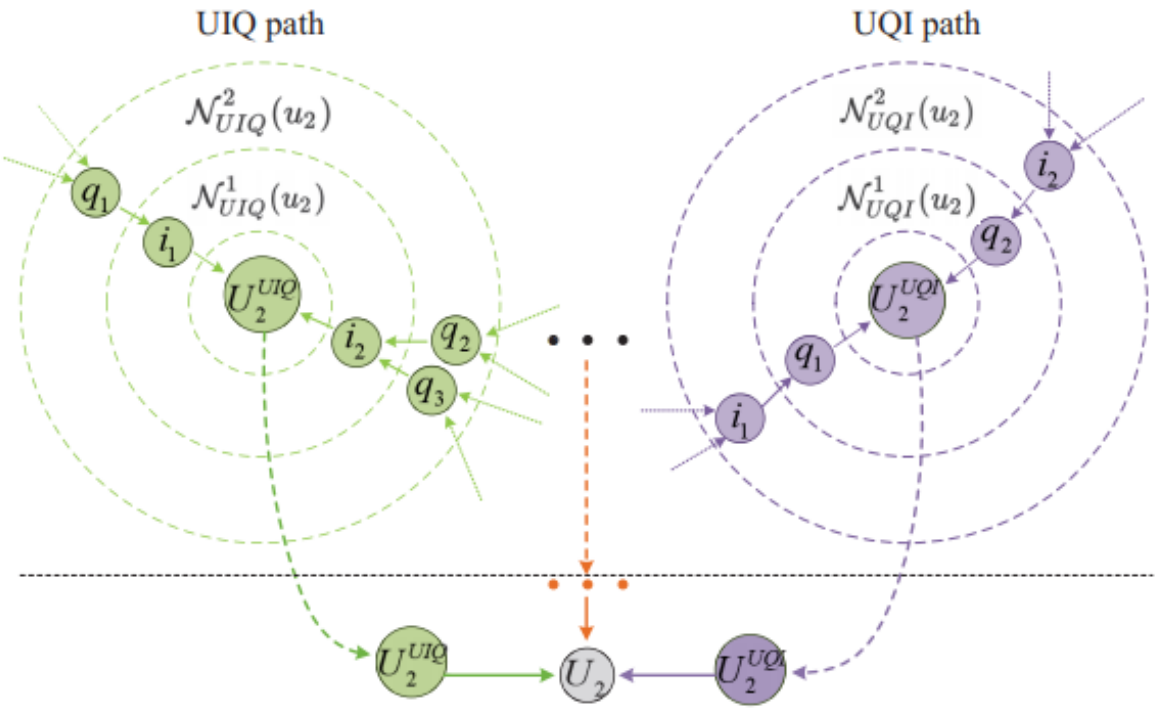


MEIRec Method



The framework of MEIRec

Metapath-guided Neighbor Aggregation



- Fuse heterogeneous information
  - Leverage metapaths to obtain different-step neighbors of an object
  - Different aggregation functions are designed for different types of neighboring information
  - More information can be added by expanding metapaths

Offline experiments

Method	1-day				3-day				5-day			
	40%	60%	80%	100%	40%	60%	80%	100%	40%	60%	80%	100%
NeuMF	0.6014	0.6066	0.6136	0.6143	0.6168	0.6218	0.6249	0.6291	0.6172	0.6224	0.6246	0.6295
LR	0.6854	0.6838	0.6884	0.6889	0.6844	0.6863	0.6857	0.6865	0.6817	0.6831	0.6827	0.6836
LR+DW	0.6878	0.6904	0.6898	0.6930	0.6888	0.6896	0.6898	0.6900	0.6838	0.6842	0.6863	0.6867
LR+MP	0.6918	0.6936	0.6950	0.6969	0.6919	0.6930	0.6933	0.6933	0.6874	0.6890	0.6898	0.6899
DNN	0.6939	0.6981	0.6991	0.6997	0.6966	0.6985	0.6999	0.7008	0.6996	0.7011	0.7017	0.7029
DNN+DW	0.6962	0.6980	0.7003	0.7024	0.7005	0.7017	0.7024	0.7030	0.7017	0.7029	0.7040	0.7047
DNN+MP	0.6984	0.6992	0.7024	0.7057	0.7025	0.7040	0.7051	0.7057	0.7017	0.7044	0.7060	0.7069
GBDT	0.7071	0.7071	0.7067	0.7073	0.7070	0.7071	0.7072	0.7071	0.7067	0.7068	0.7072	0.7066
GBDT+DW	0.7114	0.7119	0.7112*	0.7118*	0.7109	0.7106	0.7106	0.7104	0.7109	0.7112	0.7109	0.7114
GBDT+MP	0.7122*	0.7127*	0.7110	0.7111	0.7123*	0.7122*	0.7122*	0.7124*	0.7118*	0.7114*	0.7114*	0.7120*
MEIRec	**0.7273**	**0.7302**	**0.7339**	**0.7346**	**0.7352**	**0.7369**	**0.7380**	**0.7390**	**0.7372**	**0.7401**	**0.7409**	**0.7425**
Improvement	2.1%	2.5%	3.2%	3.2%	3.2%	3.5%	3.6%	3.7%	3.6%	4.0%	4.1%	4.3%

MEIRec significantly outperforms GBDT, DNN, and MF based methods

Online experiments

Table 3: Online A/B testing experiments results.

Data	Methods	CTR	Unique Click	UCTR
Android	GBDT	1.746%	256,116	13.939%
	MEIRec	1.758%	260,634	14.229%
	Improvement	0.70%	1.76%	2.07%
IOS	GBDT	0.7687%	62,462	5.2579%
	MEIRec	0.8056%	65,895	5.5436%
	Improvement	4.79%	5.50%	5.43%
Total	GBDT	1.4035%	318,578	10.5252%
	MEIRec	1.4252%	326,529	10.8052%
	Improvement	1.54%	2.50%	2.66%

MEIRec significantly improves key metrics considered by the platform and attracts more new users to search the recommended query