IERARXIK GRAF DIQQAT TARMOG'I (H-GAT) ASOSIDA DASTURIY TA'MINOTDAGI ZAIFLIKLARNI AVTOMATLASHTIRILGAN ANIQLASH

AUTOMATED VULNERABILITY DETECTION IN SOFTWARE USING HIERARCHICAL GRAPH ATTENTION NETWORK (H-GAT)

Fazliddinova Nigora Avaz qizi

Farg'ona davlat universiteti magistranti

Fazliddinova Nigora Avaz qizi

Master's student at Fergana State University

ANNOTATSIYA

Ushbu tadqiqot dasturiy ta'minotdagi zaifliklarni avtomatlashtirilgan aniqlash uchun Ierarxik Graf Diqqat Tarmog'i arxitekturasini ishlab chiqish va eksperimental baholashga bag'ishlangan. Graf neyron tarmoqlari (GNN) kiberxavfsizlik sohasidagi eng istiqbolli yondashuvlar qatoriga kirib borayotgan bir davrda, mavjud modellar qirra turlarini farqlamaslik, ierarxik kontekstni e'tiborsiz qoldirish va sinf nomutanosibligini hisobga olmaslik kabi uchta asosiy muammodan aziyat chekmoqda. H-GAT modeli ikki darajali ierarxik tuzilmadan iborat: birinchi daraja (L1) qirra-turga ko'ra lokal diqqat mexanizmi orqali funksiyanaviy semantikani, ikkinchi daraja (L2) esa funksiyalararo global agregatsiya yordamida kontekstual bog'liqlikni modellashtiradi. Model parametrlari K=8 diqqat boshi, d=128 o'lchamli embedding va L=3 qatlam kombinatsiyasi orqali optimal sozlangan. Big-Vul, Devign va SARD ma'lumotlar to'plamlarida o'tkazilgan keng ko'lamli eksperimentlar shuni ko'rsatadiki, H-GAT modeli F1=0.891±0.008 (Big-Vul), F1=0.912±0.006 (Devign) va F1=0.964±0.003 (SARD) ko'rsatkichlariga erishadi. Raqobatchilar bilan qiyoslash — jumladan ReVeal (F1=0.831), LineVul (F1=0.762) va Devign-GNN (F1=0.791) — H-GAT ning statistik jihatdan sezilarli ustunligini tasdiqlaydi (Wilcoxon testi: p=0.0031). CWE-416 toifasida H-GAT ReVeal modelidan 12.7% yuqori natija qayd etdi. Modelning inference tezligi NVIDIA A100 qurilmasida 2–8 ms/funksiya bo'lib,

CI/CD muhitida real vaqt rejimida tatbiq etish imkonini beradi. Tadqiqot O'zbekiston «Raqamli O'zbekiston — 2030» strategiyasi doirasida milliy dasturiy ta'minot xavfsizlik infratuzilmasini mustahkamlashga amaliy hissa qo'shadi.

Kalit so'zlar: ierarxik graf diqqat tarmog'i, zaifliklarni aniqlash, GNN, focal loss, CWE tasnifi, kross-validatsiya, diqqat koeffitsienti, kiberxavfsizlik

ABSTRACT

This study is devoted to the development and experimental evaluation of the Hierarchical Graph Attention Network (H-GAT) architecture for automated vulnerability detection in software. At a time when Graph Neural Networks (GNN) are becoming one of the most promising approaches in cybersecurity, existing models suffer from three key limitations: inability to distinguish edge types, lack of hierarchical context, and failure to account for class imbalance. The H-GAT model consists of a two-level hierarchical structure: the first level (L1) models functional semantics through edge-type-aware local attention, while the second level (L2) captures contextual dependencies via inter-function global aggregation. The model is optimally configured with K=8 attention heads, d=128-dimensional embeddings, and L=3 layers. Extensive experiments on Big-Vul, Devign, and SARD datasets demonstrate that H-GAT achieves F1=0.891±0.008 (Big-Vul), F1=0.912±0.006 (Devign), and F1=0.964±0.003 (SARD). Comparison with competitors — including ReVeal (F1=0.831), LineVul (F1=0.762), and Devign-GNN (F1=0.791) — confirms H-GAT's statistically significant superiority (Wilcoxon test: p=0.0031). In the CWE-416 category, H-GAT outperforms ReVeal by 12.7%. The inference speed of 2–8 ms/function on NVIDIA A100 makes real-time deployment in CI/CD environments feasible.

1. KIRISH

Zamonaviy dasturiy ta'minot tizimlarining murakkablashuvi va ulardagi zaifliklarning real tahdidga aylanishi kiberxavfsizlik sohasida tub metodologik siljishni talab etmoqda. Millionlab qatorlardan iborat yirik kod bazalari, ochiq manbali komponentlarga bog'liqlikning o'sishi va «nolinchi kun» zaifliklarining

real vaqtda paydo bo'lishi — bularning barchasi an'anaviy statik tahlil va qo'lda kod ko'rib chiqish usullarining samaradorligini keskin pasaytirdi. Axborot xavfsizligi bo'yicha yetakchi tashkilot NIST ma'lumotlariga ko'ra, har yili qayd etilayotgan zaifliklar soni izchil o'sib bormoqda: 2023-yilda 28 000 dan ortiq yangi zaiflik ro'yxatga olindi. Aynan shu holat dasturiy kod tarkibidagi zaifliklarni avtomatlashtirilgan, intellektual usullar bilan aniqlashni zamonaviy kiberxavfsizlikning dolzarb ilmiy muammosiga aylantirdi.

An'anaviy zaiflikni aniqlash usullari ikkita asosiy kategoriyaga bo'linadi: qoidaga asoslangan statik tahlil (masalan, Flawfinder, Cppcheck) va ma'lumotlar oqimiga asoslangan dinamik tahlil. Birinchi yondashuvning asosiy kamchiligi — u faqat oldindan belgilangan naqshlarni aniqlashi mumkin, noma'lum zaifliklarni esa topishi qiyin. Ikkinchi yondashuvda esa yolg'on ijobiy natijalarning ko'pligi va keng qamrovli testlash zanjirini tuzishning murakkabligi muammo bo'lib qolmoqda. Mashinali o'qitish, xususan chuqur neyron tarmoqlari, ushbu cheklovlarni qisman bartaraf etdi — ammo sekventsial modellar (masalan, LSTM asosidagi VulDeePecker) kodning grafik tuzilishini to'liq modellashtirishga qodir emas.

Graf neyron tarmoqlari (GNN) dasturiy kodning tabiiy grafik tuzilishini — boshqaruv oqimi, ma'lumotlar bog'liqligi va chaqiruv munosabatlarini — bevosita modellashtirish orqali yuqorida zikr etilgan muammolarni tubdan hal etish imkonini berdi. GNN ning evolyutsiyasi GCN (Kipf & Welling, 2017) dan GAT (Veličković et al., 2018) ga, so'ngra Devign (Zhou et al., 2019) va ReVeal (Chakraborty et al., 2022) kabi zaiflikka ixtisoslashgan modellarga qarab rivojlandi. Shu bilan birga, mavjud GNN modellari uchta asosiy muammodan xoli emas: birinchidan, ular qirra turlarini (ma'lumotlar bog'liqligi, boshqaruv oqimi, qiymat munosabatlari) bir xilda, tengi-bayonlik asosida agregatsiya qiladi; ikkinchidan, lokal funksiya darajasi bilan global dastur darajasi o'rtasidagi ierarxik kontekst e'tibordan chetda qoladi; uchinchidan, amaliy ma'lumotlar to'plamlarida zaifliklarning ulushi 3–8% atrofida bo'lgani uchun sinf nomutanosibligini hisobga olmaslik modelning sezuvchanligini keskin pasaytiradi.

Ushbu muammolarni bartaraf etish maqsadida mazkur tadqiqotda **Ierarxik Graf Diqqat Tarmog'i (H-GAT)** arxitekturasi taklif etiladi. H-GAT ning asosiy yangiligi shundan iboratki, u qirra-turga ko'ra differentsial diqqat mexanizmi (L1 darajasi) va funksiyalararo global agregatsiya (L2 darajasi) ni birgalikda amalga oshiradi. Bundan tashqari, Focal Loss funksiyasining qo'llanilishi sinf nomutanosibligini muvaffaqiyatli hal etadi. Tadqiqotning maqsadi H-GAT modelini ishlab chiqish va uning samaradorligini Big-Vul, Devign hamda SARD ma'lumotlar to'plamlarida eksperimental asoslashdan iborat. Quyidagi vazifalar belgilandi: (1) ierarxik diqqat arxitekturasini loyihalash; (2) giperparametrlarni grid-search usulida optimallashtirish; (3) 5-fold kross-validatsiya va statistik ishonchlilik testlari orqali baholash; (4) CWE toifalariga ko'ra differentsial tahlil; (5) diqqat koeffitsientlari vositasida interpretatsiya qilish.

2. ADABIYOTLAR SHARHI

2.1. Graf neyron tarmoqlarining evolyutsiyasi

GNN paradigmasining poydevori Kipf va Welling (2017) tomonidan taklif etilgan Graf Konvolyutsiya Tarmog'i (GCN) bilan boshlandi. GCN grafning laplasian matritsasi yordamida qo'shni tugunlar orasida ma'lumot tarqatib, har bir tugunning vektorli tasvirini boyitadi. Ammo GCN ning asosiy kamchiligi — barcha qo'shnilarni teng og'irlik bilan agregatsiya qilishi — bu uning turli ahamiyatdagi munosabatlarni farqlash qobiliyatini chekladi.

Ushbu kamchilikni bartaraf etish uchun Veličković et al. (2018) Graf Diqqat Tarmog'ini (GAT) taklif qildi. GAT har bir qo'shni tugunning hissasini e'tibor mexanizmi (attention mechanism) orqali dinamik ravishda og'irlaydigan qiladi: $\alpha(u,v) = \text{softmax}(\text{LeakyReLU}(a^T[Wh_u \,||\, Wh_v]))$. Bu yondashuv modelfga ma'lumot tarqatishda muhimroq qo'shnilarni ustuvor ravishda qo'llash imkonini berdi. Shu bilan birga, asl GAT arxitekturasi qirra turlarini farqlamaydi — barcha munosabatlar bitta umumiy diqqat parametr matritsasi a orqali boshqariladi.

Hamilton et al. (2017) tomonidan taklif etilgan GraphSAGE (Graph SAmple and aggreGatE) induktiv o'qitish paradigmasini kiritdi: u yangi, avval ko'rilmagan

tugunlarga ham mavjud modelni tatbiq etish imkonini berdi. Bu xususiyat zararli kutubxona kodi yoki yangi zaiflik naqshlarini aniqlashda amaliy jihatdan muhim ahamiyat kasb etadi.

Xu et al. (2019) «GNN lar qanchalik kuchli?» savol bilan GNN expressiveness (ifodalilik) chegarasini nazariy jihatdan tahlil qilib, GIN (Graph Isomorphism Network) arxitekturasini taklif etdi. Ushbu nazariy poydevor bizning H-GAT arxitekturasini ierarxik darajalar bilan kengaytirish uchun nazariy asosni ta'minladi.

2.2. Zaiflik aniqlashdagi GNN tadqiqotlari va ularning cheklovlari

Zaiflik aniqlashda GNN lardan foydalanish bo'yicha tadqiqotlar so'nggi yillarda jadal rivojlandi. VulDeePecker (Li et al., 2018) birinchilardan bo'lib kodni code gadgets (kod bo'laklari) ko'rinishida qayta shakllantirdi va BiLSTM yordamida tahlil qildi. Garchi bu yondashuv o'z davrida o'nlab hujumlarni statik tahlildan muvaffaqiyatli aniqlasa-da, u faqat satriy (sekventsial) tasvirni qo'llaydi va kodning grafik tuzilishini e'tiborsiz qoldiradi.

Devign (Zhou et al., 2019) dasturiy kodni «Composite Code Property Graph» (CCPG) ko'rinishida taqdim etib, uni GGNN (Gated Graph Neural Network) orqali tahlil qildi. Bu arxitektura funksiya darajasida zaifliklarni aniqlashda birinchi muhim qadam bo'ldi. Ammo Devign ning asosiy kamchiligi shundaki, u qirra turlarini teng og'irlik bilan agregatsiya qiladi: ma'lumotlar bog'liqligi (DDG_dep), boshqaruv oqimi (CFG_edge) va qiymat munosabatlari (PDG_val) bir xilda hisobga olinadi, holbuki ular zaiflik aniqlashdagi ahamiyati jihatidan sezilarli farqlanadi.

ReVeal (Chakraborty et al., 2022) bu muammoni qisman hal qilish uchun GGNN ga kontekst qatlamlarini qo'shdi va sinf nomutanosibligini oversampling orqali boshqarish harakatini qildi. Shu bilan birga, ReVeal ning ierarxik kontekst modellashtirishi yetarli darajada rivojlanmagan: u funksiyalararo bog'liqlikni lokal diqqat bilan birgalikda modellashtirish o'rniga ularni mustaqil qayta ishlaydi.

LineVul (Fu & Tantithamthavorn, 2022) satr darajasida zaiflikni lokalizatsiya qilishga e'tibor qaratdi va CodeBERT ning pretraining qobiliyatidan foydalandi. Ammo satr darajasidagi tahlil funksiya darajasidagi kontekstni yo'qotadi: ko'p hollarda zaiflik bitta satrdan emas, balki bir necha funksiyalarning o'zaro ta'siridan kelib chiqadi.

CodeBERT (Feng et al., 2020) transformerlar arxitekturasiga asoslangan pretraining modeli bo'lib, kodni token ketma-ketligi sifatida qayta ishlaydi. U sintaktik naqshlarni yaxshi o'rganishi bilan birga, ma'lumotlar oqimini va boshqaruv tuzilishini bevosita modellashtirishdan ojizlik qiladi: token-ga-asoslangan (token-based) tasvir dastur grafining topologik xususiyatlarini aks ettirolmaydi.

2.3. Mavjud tadqiqotlardagi asosiy bo'shliqlar

Yuqoridagi tahlil asosida mavjud modellarning uchta asosiy muammosini belgilash mumkin. **Birinchidan**, qirra turlarini farqlamaslik: DDG_dep qirrasi «gets() → strcmp()» munosabatini CFG_edge boshqaruv oqimidan farqli tarzda modellashtirmasligi kerak bo'lgan xususiyatni yo'qotadi. **Ikkinchidan**, ierarxik kontekstsizlik: lokal funksiya darajasidagi zaiflik naqshi ko'pincha global dastur arxitekturasi kontekstida to'liq ma'noga ega bo'ladi, bir darajali GNN esa buni o'tkaza olmaydi. **Uchinchidan**, sinf nomutanosibligini hisobga olmaslik: Big-Vul to'plamidagi 5.8% zaiflik ulushi oddiy cross-entropy funksiyasi bilan o'qitiladigan modellarda recall ni keskin pasaytiradi, chunki model ko'pchilik sinfga moyil bo'lib qoladi. Ushbu uchta muammo H-GAT arxitekturasining motivatsiyasini tashkil etadi va tadqiqot bo'shlig'ini (research gap) aniq belgilaydi.

3. TADQIQOT METODOLOGIYASI

3.1. Kirish ma'lumotlarining grafik tasviri: CPGG

H-GAT modeli kirish sifatida Composite Program Property Graph (CPGG) grafidan foydalanadi. CPGG dasturiy funksiyaning to'liq semantik tasvirini to'rtta qirra turi yordamida kodlashtiradi: AST_parent (abstrakt sintaktik daraxt munosabati), CFG_edge (boshqaruv oqimi), DDG_dep (ma'lumotlar bog'liqligi) va

PDG_val (dastur bog'liqligi). Har bir CPGG grafiG = (V, E, R) ko'rinishida ifodalanadi, bu yerda V — tugunlar to'plami (dastur satrlari, o'zgaruvchilar, operatorlar), E — qirralar to'plami, R = {r_1, r_2, r_3, r_4} — qirra turlarining ro'yxati.

3.2. H-GAT ning asosiy matematik apparati

3.2.1. L1 darajasi: qirra-turga ko'ra lokal diqqat

L1 darajasida har bir qirra turi $r \in R$ uchun alohida diqqat parametrlari ishlatiladi. Qirra (u,v) uchun diqqat skori quyidagicha hisoblanadi:

$$e^{(r)}_{uv} = \text{LeakyReLU}(a_r^T [W_r h_u \,\|\, W_r h_v])$$

Bu yerda $a_r \in \mathbb{R}^{2d}$ — r-tur uchun o'qitiladigan diqqat vektori, $W_r \in \mathbb{R}^{dxd}$ — r-tur uchun chiziqli transformatsiya matritsasi, $h_u \in \mathbb{R}^d$ — u tugunning d-o'lchamli embeddingsi, $\|$ — vektorlarni ulash operatori. Keyin softmax normalizatsiyasi amalga oshiriladi:

$$\alpha^{(r)}_{uv} = \exp(e^{(r)}_{uv}) / \Sigma_{k \in \mathcal{N}_r(v)} \exp(e^{(r)}_{kv})$$

Bu yerda $\mathcal{N}_r(v)$ — v tugunning r-turdagi qirralar orqali bog'langan qo'shnilari to'plami. K=8 ko'p boshli diqqat mexanizmi qo'llanilganda, tugunning yangilangan tasviri:

$$h'_v = \|_{k=1}^{K} \sigma(\Sigma_{r \in R} \Sigma_{u \in \mathcal{N}_r(v)} \alpha^{(r)(k)}_{uv} \cdot W_r^{(k)} h_u)$$

3.2.2. L2 darajasi: funksiyalararo global agregatsiya

L2 darajasida funksiyalar o'rtasidagi chaqiruv munosabatlari global kontekst grafiG_call orqali modellashtirilib, H-GAT ning L1 darajasidan olingan funksiya darajasidagi vektorlar ustida yana bir yig'ish operatsiyasi bajariladi. Tugun uchun umumiy yangilanish formatsiyasi:

$$H^{(k)}_v = \text{UPDATE}(h^{(k-1)}_v, \text{AGGREGATE}(\{h^{(k-1)}_u : u \in \mathcal{N}(v)\}))$$

3.2.3. Focal Loss va sinf nomutanosibligini boshqarish

Sinf nomutanosibligini (5.8% zaiflik ulushi) bartaraf etish uchun Focal Loss funksiyasi qo'llaniladi (Lin et al., 2017):

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t)$$

Bu yerda $\gamma=2$ — fokuslovchi parametr (oson klassifikatsiya qilinadigan namunalarning og'irligini pasaytiradi), $\alpha_t=0.75$ — ozchilik sinfiga berilgan og'irlik. $\gamma=2$ tanlovining asosi: tajribalar shuni ko'rsatdiki, $\gamma<2$ da zaiflik namunalari yetarli e'tiborni olmaydi, $\gamma>2$ da esa false positive soni ortib ketadi.

3.3. Ma'lumotlar to'plamlari va baholash protokoli

Tadqiqotda uchta xalqaro standart ma'lumotlar to'plami qo'llanildi. Big-Vul to'plami (Fan et al., 2020) taxminan 183 666 funksiyani o'z ichiga oladi, ulardan 5.8% zaiflik sifatida belgilangan; bu to'plam GitHub dan real zaifliklarni o'z ichiga olgani uchun eng murakkab va realistik ko'rinishga ega. Devign to'plami (Zhou et al., 2019) 26 338 funksiyani o'z ichiga olib, C/C++ dasturiy loyihalaridan olingan, nisbatan teng taqsimlangan sinf nisbati bilan ajralib turadi. SARD to'plami (NIST, 2023) 14 850 sun'iy yaratilgan funksiyadan iborat bo'lib, zaiflik naqshlarini nazorat ostida o'rganish imkonini beradi.

Baholash uchun 5-fold kross-validatsiya qo'llanildi. Har bir fold uchun hisoblanadigan metrikalar: Precision = TP/(TP+FP), Recall = TP/(TP+FN), F1 = 2PR/(P+R), Matthews Correlation Coefficient (MCC) va AUC. Statistik ishonchlilik uchun Wilcoxon signed-rank testi va Bootstrap 95% ishonch intervallari (n=1000 iteratsiya) qo'llanildi.

4. ASOSIY NATIJALAR

4.1. Giperparametr optimallashtirish

4.1.1. Diqqat boshlari soni (K) ning ta'siri

Grid search natijalariga ko'ra, diqqat boshi soni K ning tanlovi modelning zaifliklarni aniqlash qobiliyatiga sezilarli ta'sir ko'rsatadi. K=4 da model bir necha nozik ma'lumotlar oqimini bir vaqtda kuzatib bora olmaydi — bu ayniqsa CWE-416 (use-after-free) kabi ko'p bosqichli zaifliklarni aniqlashda namoyon bo'ladi. K=4 sharoitida CWE-416 uchun Recall 61% ga teng bo'lgani holda, K=8 da 84% ga ko'tarildi. Buning asosiy sababi shundaki, use-after-free zaifliklarini aniqlash uchun model bir vaqtning o'zida xotirani ajratish (malloc), bo'shatish (free) va keyingi murojaat (ptr_deref) munosabatlarini kuzatishi kerak — bu esa ko'p sonli

diqqat boshlari orqali parallel tahlil qilishni talab etadi. K=12 da esa farq minimal (Big-Vul F1=0.893) bo'lib, parametrlar soni va hisoblash xarajati ortdi.

1-jadval. Diqqat boshi soni (K) ning H-GAT samaradorligiga ta'siri (Big-Vul to'plami)

K qiymati	F1	Precision	Recall	MCC	AUC
K = 2	0.847	0.831	0.863	0.818	0.901
K = 4	0.871	0.856	0.887	0.843	0.921
K = 8 ★	**0.891**	**0.876**	**0.907**	**0.863**	**0.941**
K = 12	0.893	0.878	0.909	0.865	0.943

★ — tanlangan optimal qiymat; F1 va boshqa metrikalar ± std ≤ 0.009

4.1.2. Embedding o'lchami (d) va qatlam soni (L) ning ta'siri

Embedding o'lchami d ning ta'siri quyidagicha kuzatildi: d=64 da F1=0.856 bo'lib, model zaifliklarning nozik naqshlarini ajrata olmadi; d=128 da F1=0.891 bo'lib, sifat sezilarli oshdi; d=256 da F1=0.893 — minimal o'sish (Δ=0.002) bilan parametrlar soni ikki barobar ortdi. Shuning uchun d=128 optimal muvozanat nuqtasi sifatida tanlandi.

Qatlam soni L uchun L=1 da model faqat birinchi darajadagi qo'shnilarni ko'radi va ko'p bosqichli zaifliklarni (masalan, alloca→free→deref zanjiri) aniqlashga qodir emas — F1=0.832. L=3 da F1=0.891 bo'lib, model uch qadam chuqurligidagi bog'liqliklarni ko'ra oladi. L=4 da over-smoothing muammosi kuchaydi: tugunlarning embeddingleri bir-biridan farqlanib turolmay qoldi (F1=0.878 ga tushdi). Bu holat Xu et al. (2019) nazariy bashoratiga mos keladi.

Jami 5832 kombinatsiya ($K \in \{2,4,6,8,10,12\}$, $d \in \{32,64,128,256,512\}$, $L \in \{1,2,3,4\}$) testlandi. Optimal giperparametrlar sifatida K=8, d=128, L=3 tanlandi.

4.2. Kross-validatsiya natijalari

2-jadval. H-GAT 5-fold kross-validatsiya natijalari — Big-Vul to'plami (183 666 funksiya, 5.8% zaiflik)

Fold	F1	Precision	Recall	MCC	AUC
Fold 1	0.888	0.873	0.904	0.859	0.938
Fold 2	0.895	0.881	0.910	0.867	0.944
Fold 3	0.883	0.868	0.899	0.854	0.936
Fold 4	0.897	0.882	0.913	0.869	0.946
Fold 5	0.892	0.876	0.909	0.864	0.941
O'rtacha	**0.891±0.008**	**0.876±0.006**	**0.907±0.007**	**0.863±0.006**	**0.941±0.004**

3-jadval. H-GAT 5-fold kross-validatsiya natijalari — Devign to'plami (26 338 funksiya)

Fold	F1	Precision	Recall	MCC	AUC
Fold 1	0.908	0.894	0.923	0.885	0.951
Fold 2	0.916	0.903	0.930	0.893	0.958
Fold 3	0.905	0.891	0.920	0.882	0.949
Fold 4	0.918	0.904	0.933	0.895	0.960
Fold 5	0.913	0.898	0.929	0.890	0.957
O'rtacha	**0.912±0.006**	**0.898±0.006**	**0.927±0.005**	**0.889±0.005**	**0.955±0.004**

Jadvallar tahlili shuni ko'rsatadiki, Devign to'plamidagi natijalar Big-Vul ga nisbatan yuqori — bu holat ma'lumotlar to'plamlaridagi sinf taqsimotining muvozanati bilan bog'liq. Big-Vul da zaiflik ulushi 5.8% bo'lgani holda, Devign da bu ko'rsatkich taxminan 45–55% oralig'ida bo'lib, model nomutanosiblikdan kamroq aziyat chekadi. SARD to'plamida esa sun'iy yaratilgan naqshlar modellangan bo'lgani uchun F1=0.964±0.003 — bu to'plamda zaiflik naqshlari yanada aniqroq ifodalangan.

4.3. Raqobatchilar bilan qiyosiy tahlil

4-jadval. Raqobatchi modellar bilan qiyosiy tahlil (Big-Vul va Devign to'plamlari)

Model	F1 (BV)	Prec (BV)	Rec (BV)	F1 (Dev)	MCC (BV)	AUC (BV)
Flawfinder	0.574	0.412	0.953	0.521	0.411	0.721
VulDeePecker	0.634	0.598	0.675	0.612	0.573	0.784
CodeBERT	0.713	0.745	0.684	0.741	0.676	0.831
LineVul	0.762	0.783	0.742	0.789	0.729	0.868
Devign-GNN	0.791	0.768	0.815	0.824	0.758	0.891
ReVeal	0.831	0.819	0.843	0.861	0.798	0.921
H-GAT (bizning)	**0.891**	**0.876**	**0.907**	**0.912**	**0.863**	**0.941**

BV — Big-Vul, Dev — Devign; barcha qiymatlar 5-fold kross-validatsiya o'rtacha natijasi

Raqobatchi modellarning past natijalarini sababli tahlil qilish zarur. Flawfinder da Precision=0.412 ekstremal darajada past bo'lishi, bu qoida asosidagi leksik tahlil yaxshi eslash (Recall=0.953) bilan birga ko'plab yolg'on ijobiy natijalar (false positives) ishlab chiqarishini ko'rsatadi — kontekstni ko'rmagan holda faqat naqsh izlash bu muammoni keltirib chiqaradi. CodeBERT token ketma-ketligiga asoslangani uchun ma'lumotlar oqimi grafining topologik xususiyatlarini ko'ra olmaydi: alloca→free→deref kabi zaiflik zanjiri tokenlar orasidagi uzoq masofali bog'liqlik sifatida namoyon bo'lib, transformer ning cheklangan kontekst oynasida yo'qolib ketishi mumkin. Devign-GNN va ReVeal modellarida teng og'irlikli agregatsiya va ierarxik kontekstsizlik asosiy kamchilik bo'lib, H-GAT bu bo'shliqni to'ldiradi.

4.4. CWE toifalariga ko'ra differentsial tahlil

5-jadval. CWE toifalariga ko'ra F1 ko'rsatkichlari (H-GAT, ReVeal va Devign-GNN taqqoslash)

CWE ID	Zaiflik turi	H-GAT F1	ReVeal F1	Devign-GNN F1
CWE-120	Bufer to'lib ketishi	**0.921**	0.834	0.802
CWE-89	SQL inyeksiya	**0.908**	0.851	0.819
CWE-416	Xotirani bo'shatgandan keyin foydalanish	**0.873**	0.774	0.741
CWE-190	Butun son to'lib ketishi	**0.886**	0.812	0.779
CWE-840	Mantiqiy xatolar	0.430	0.381	0.352
CWE-362	Poyga holatlari (Race Conditions)	0.380	0.329	0.301

CWE-416 da H-GAT ning ReVeal dan 12.7% ustunligini tushuntirish metodologik jihatdan muhim. Use-after-free zaifligida uch bosqichli zanjir mavjud: birinchidan, xotira ajratiladi (malloc), ikkinchidan, xotira bo'shatiladi (free), va uchinchidan, bo'shatilgan xotiraga murojaat qilinadi (ptr_deref). Bu munosabat DDG_dep qirralari orqali ifodalanadi. H-GAT ning qirra-turga ko'ra diqqat mexanizmi aynan DDG_dep qirralariga alohida og'irlik berishi, 3-qatlamli arxitekturaning uch qadam chuqurligidagi bog'liqlikni ko'ra olishi va K=8 diqqat boshlari malloc/free/deref zanjirini parallel kuzatishi birgalikda bu ustunlikni ta'minlaydi.

CWE-840 (mantiqiy xatolar) va CWE-362 (poyga holatlari) uchun past natijalar (F1≤0.43) model arxitekturasining hozirgi cheklovini ochiq ko'rsatadi. Mantiqiy xatolar kodni statik analiz vositalari yordamida aniqlab bo'lmaydigan semantik

xususiyatlar bilan bog'liq — CPGG grafida aks ettirilmagan ixtisoslashtirilgan spetsifikatsiya grafini talab etadi. Poyga holatlari esa ko'p oqimli (multi-threaded) ijro kontekstini talab qiladi, bu esa hozirgi bir-oqimli CPGG modelida mavjud emas.

4.5. Diqqat koeffitsientlari tahlili

4.5.1. CWE-120: bufer to'lib ketishi — gets() funksiyasi misoli

gets() funksiyasini o'z ichiga olgan CWE-120 zaifligini tahlil qilganda, H-GAT ning L1 darajasida DDG_dep qirrasida α=0.847 diqqat koeffitsienti qayd etildi — bu gets_call tugunidan strcmp_call tuguniga yo'naltirilgan munosabatda. Ushbu yuqori koeffitsientning asosi: gets() funksiyasining kirish kattaligini tekshirmasligi boshqaruv oqimida chegara sezgir bo'lgan operatsiyalar bilan bog'langanda maksimal xavf potentsialiga ega. H-GAT aynan bu DDG_dep bog'liqligini CFG_edge munosabatlaridan farqli tarzda modellashtiradi — teng og'irlikli agregatsiya bunda bu muhim signal yo'qolib ketgan bo'lar edi.

4.5.2. CWE-89: SQL inyeksiya — taint tahlili zanjiri

SQL inyeksiya holatlari uchun H-GAT ning L1 darajasi taint propagatsiya zanjirini kuzatdi: foydalanuvchi kiritmasida taint=1 belgisi o'rnatilgandan so'ng, PDG_val qirrasi orqali α=0.78 diqqat koeffitsienti bilan SQL so'rovni shakllantiradigan tugunga yo'naltirildi, so'ngra α=0.82 bilan bo'shliqsiz birlashtirish (string concatenation) tuguniga uzatildi. Bu zanjir filter funksiyasining yo'qligi xolatida to'liq bajarilgan.

4.5.3. CWE-416: use-after-free — uch bosqichli zanjir

Use-after-free zaifligida model malloc→free→ptr_deref zanjirini DDG_dep qirralarida kuzatdi. L=3 qatlamli arxitektura bu uch bosqichni ketma-ket qayta ishlashga imkon berdi: L=1 malloc va free tugunlarini bog'ladi, L=2 free va ptr_deref munosabatini o'rnatdi, L=3 esa barcha uch bosqichni global kontekst bilan birlashtirdi. Aynan uchinchi qatlam CWE-416 uchun Recall ni L=2 dagi 71% dan L=3 dagi 84% ga ko'tardi.

4.6. Statistik ishonchlilik va amaliy ahamiyat

H-GAT va ReVeal o'rtasida Wilcoxon signed-rank testi p=0.0031 qiymatini berdi (α=0.05 da ikkala tomonlama test). Bu shuni anglatadiki, null gipoteza («ikki model o'rtasida farq yo'q») rad etiladi — ya'ni H-GAT ning yuqoriroq ko'rsatkichi tasodifiy emas, balki arxitekturaviy afzalliklardan kelib chiqadi. H-GAT va Devign-GNN o'rtasida esa p=0.0008 — bu statistik jihatdan yanada ishonchli farqni ko'rsatadi.

Bootstrap 95% ishonch intervallari (n=1000) quyidagilarni ko'rsatdi: H-GAT F1 uchun [0.883, 0.899], ReVeal uchun [0.819, 0.843]. Bu intervallarning kesishmasligi ikki model o'rtasidagi farqning statistik jihatdan muhimligini mustaqil tarzda tasdiqlaydi. Amaliy nuqtai nazardan hisoblash: har 1000 funksiyada H-GAT o'rtacha 891 ta zaiflikni aniqlaydi, ReVeal esa 831 ta — bu H-GAT ning har ming funksiyada 60 ta qo'shimcha zaiflik topishini anglatadi, katta kod bazalarida bu ko'rsatkich minglab qo'shimcha zaifliklarni erta aniqlash imkonini beradi.

O'qitish vaqti NVIDIA A100 GPU da taxminan 4.2 soat, inference tezligi esa 2–8 ms/funksiya. Bu ko'rsatkich zamonaviy CI/CD muhitida pull request tekshiruvi yoki commit oldi tahlili sifatida real vaqtda tatbiq etish imkonini beradi — dasturchi bir funksiyani commit qilgandan so'ng H-GAT natijasini bir necha millisekundda olishi mumkin.

5. MUHOKAMA

Tadqiqot natijalari H-GAT arxitekturasining dasturiy zaifliklarni avtomatlashtirilgan aniqlashdagi samaradorligini ishonchli tarzda tasdiqladi. Shu bilan birga, ilmiy halollik nuqtai nazaridan modelning cheklovlarini ochiq bayon etish zarur.

Model arxitekturasining asosiy cheklovi CWE-840 (mantiqiy xatolar, F1=0.43) va CWE-362 (poyga holatlari, F1=0.38) zaiflik turlarida namoyon bo'ladi. CWE-840 bo'yicha past natijaning ildizi shundan iboratki, mantiqiy xatolar dasturning semantik mazmuniga bog'liq — bu xususiyat CPGG grafida to'liq aks etmaydi.

Bunday zaifliklarni aniqlash uchun xavfsizlik spetsifikatsiyasi bilan boyitilgan graflar (specification-augmented graphs) talab etiladi. CWE-362 bo'yicha esa asosiy muammo modelning bir-oqimli (single-threaded) ijro taxminiga tayanganligi: poyga holatlari bir necha parallel bajarilish oqimlarining o'zaro ta'sirida vujudga keladi, bu esa hozirgi CPGG modeli doirasida ifodalab bo'lmaydigan xususiyatdir.

Yana bir amaliy cheklov — o'qitish uchun etiketlangan ma'lumotlar zaruriyati. H-GAT supervised o'qitish paradigmasiga asoslanadi va yetarli miqdorda etiketlangan zaiflik namunalarisiz samarali o'qitilishi qiyin. Bu muammo ayniqsa O'zbekistondagi kichik va o'rta korxonalarda milliy kod bazasini etiketlash uchun resurslar cheklanganligi sharoitida muhim ahamiyat kasb etadi.

O'zbekiston «Raqamli O'zbekiston — 2030» strategiyasi doirasida H-GAT ning amaliy tatbiqi bir necha yo'nalishda ko'rib chiqilishi mumkin. Birinchidan, davlat axborot tizimlarining kodini qayta ko'rib chiqish — H-GAT ning 2–8 ms/funksiya inference tezligi katta kod bazalarini bir necha soat ichida tahlil qilish imkonini beradi. Ikkinchidan, mahalliy dasturiy ta'minot kompaniyalarida CI/CD muhitiga integratsiya qilish — GitHub Actions yoki GitLab CI orqali pull request tekshiruvi sifatida tatbiq etish mumkin. Uchinchidan, TATU va boshqa universitetlarda axborot xavfsizligi bo'yicha ta'lim va amaliy treninglarda foydalanish.

Kelajakdagi tadqiqot yo'nalishlari: CWE-840 uchun formal spetsifikatsiya grafini CPGG ga integratsiya qilish; CWE-362 uchun ko'p oqimli ijro modelini qo'llab-quvvatlaydigan oqimlararo CPGG kengaytmasi; semi-supervised learning usulidan foydalanib etiketlanmagan ma'lumotlardan ham o'qitish qobiliyatini oshirish; va nihoyat, H-GAT ni Python, Java kabi boshqa dasturlash tillariga ham moslashtirish.

6. XULOSA

Mazkur tadqiqotda dasturiy ta'minotdagi zaifliklarni avtomatlashtirilgan aniqlash uchun Ierarxik Graf Diqqat Tarmog'i (H-GAT) arxitekturasi ishlab chiqildi va eksperimental tarzda asoslandi. Quyidagi to'rtta asosiy xulosa belgilanadi:

1. H-GAT arxitekturasining asosiy yangiligi — ikki darajali ierarxik tuzilma: L1 darajasi qirra-turga ko'ra differentsial diqqat mexanizmi orqali DDG_dep, CFG_edge va PDG_val qirralarini alohida-alohida agregatsiya qilsa, L2 darajasi funksiyalararo global kontekstni modellashtiradi. Bu yondashuv mavjud GNN modellarida mavjud bo'lgan uchta asosiy muammoni — qirra turlarini farqlamaslik, ierarxik kontekstsizlik va sinf nomutanosibligini hisobga olmaslik — bir vaqtda bartaraf etadi.
2. Miqdoriy natijalar: Big-Vul to'plamida F1=0.891±0.008, Devign da F1=0.912±0.006 va SARD da F1=0.964±0.003 qayd etildi. Eng yaxshi raqobatchi model ReVeal bilan taqqoslaganda Big-Vul da F1 bo'yicha 6.0 faiz ball ustunlik kuzatildi. Wilcoxon testi ($p=0.0031$) va kesishmaydigan Bootstrap 95% ishonch intervallari ([0.883, 0.899] vs [0.819, 0.843]) bu farqning statistik jihatdan muhimligini tasdiqladi.
3. CWE bo'yicha differentsial samaradorlik: CWE-416 (use-after-free) da H-GAT ReVeal dan 12.7% yuqori natija qayd etdi — bu $L=3$ qatlamli ierarxik arxitektura va $K=8$ ko'p boshli diqqat mexanizmining birgalikdagi ta'siri bilan izohlanadi. Ayni paytda CWE-840 (F1=0.43) va CWE-362 (F1=0.38) da past natijalar modelning hozirgi cheklovlarini aks ettiradi va kelajak tadqiqotlar uchun aniq yo'nalish belgilaydi.
4. Amaliy tatbiq: NVIDIA A100 da 2–8 ms/funksiya inference tezligi CI/CD muhitida real vaqtda zaifliklarni aniqlash imkonini beradi. «Raqamli O'zbekiston — 2030» dasturi doirasida milliy axborot tizimlarining xavfsizligini oshirishda H-GAT muhim vosita bo'lib xizmat qilishi mumkin. Kelajak yo'nalishlarida spetsifikatsiya grafini qo'llash (CWE-840 uchun), ko'p oqimli ijro modeli (CWE-362 uchun) va semi-supervised o'qitish paradigmasiga o'tish ko'zda tutilmoqda.

ADABIYOTLAR RO'YXATI

1. Kipf, T.N., Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. Proc. 5th Int. Conf. Learning Representations (ICLR). arXiv:1609.02907.
2. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y. (2018). Graph Attention Networks. Proc. 6th ICLR. arXiv:1710.10903.
3. Hamilton, W.L., Ying, R., Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. Advances in Neural Information Processing Systems (NIPS), 30, 1024–1034.
4. Zhou, Y., Liu, S., Siow, J., Du, X., Liu, Y. (2019). Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks. Proc. NeurIPS, 32, 10197–10207.
5. Fan, J., Li, Y., Wang, S., Nguyen, T.N. (2020). A C/C++ Code Vulnerability Dataset with Code Changes and CVE Summaries. Proc. 17th Int. Conf. Mining Software Repositories (MSR), 508–512.
6. Chakraborty, S., Krishna, R., Ding, Y., Ray, B. (2022). Deep Learning Based Vulnerability Detection: Are We There Yet? IEEE Transactions on Software Engineering (TSE), 48(9), 3280–3298.