

Классын алдаанд өртөмтгий байдлыг объект хандлагат зохиомжийн хэмжигдэхүүн ашиглан таамаглах нь

Ц.Лхамролом, Б.Батням, Н.Оюун-Эрдэнэ,
Мэдээлэл, компьютерийн ухааны тэнхим
Хэрэглээний Шинжлэх ухаан Инженерчлэлийн Сургууль
Монгол Улсын Их Сургууль
(lhamrolom, batnyam, oyunerdene)@seas.num.edu.mn

Abstract— Програм хангамж хөгжүүлж байх явцад програмд гарч болох алдааг аль болох эрт үед илрүүлэн засах шаардлагатай. Програм хангамжийн чанарыг програм хангамж хөгжүүлэлтийн эрт үед тодорхойлох нэг арга нь зохиомжийн хэмжигдэхүүн ашиглан таамаглах юм. Энэхүү судалгааны ажлаар классын алдаанд өртөмтгий байдлыг зохиомжийн хэмжигдэхүүн ашиглан таамаглах загвар боловсруулахыг зорьсон. Энэ нь програмын алдааг код бичсэний дараа засах бус зохиомжийн алдааг эрт илрүүлэн зассанаар хэрэгжүүлэлтийн үед гарах алдааны цар хүрээ багасаж, хүчин чармайлтыг багасгах ач холбогдолтой.

Тус зорилгодоо хүрэхийн тулд эхлээд класс диаграмаас тодорхойлох боломжтой зохиомжийн хэмжигдэхүүнийг сонгож авсан бөгөөд тус хэмжигдэхүүн болон классын алдаанд өртөмтгий байдлын хооронд хамаарал байгаа эсэхийг тодорхойлсон. Үүний дараа машин сургалтын 6-н төрлийн алгоритм сонгон авч классын алдаанд өртөмтгий байдлыг таамаглах туршилт хийсэн. Туршилтанд нээлттэй эхийн 4 системд гарсан алдааны бүртгэлийн мэдээллийг Promise Data сангаас авч ашигласан. Туршилтын үр дүнд Random Forest арга нь хамгийн өндөр үр дүнтэй буюу F-measure нь ойролцоогоор 90%, нийт алдаатай классын 75%-ийг зөв таамагласан үзүүлэлт гарсан.

Түлхүүр үгс— зохиомжийн хэмжигдэхүүн, классын алдаанд өртөмтгий байдал, машин сургалтын арга

I. УДИРТГАЛ

A. Үндэслэл

Програм хангамж хөгжүүлэлтэд зарцуулагдах нийт зардлын ойролцоогоор 75% нь засвар үйлчилгээний зардал байдаг [1]. Програм хангамжийн засвар үйлчилгээ гэдэг нь програм хангамжийн бүтээгдэхүүнийг хэрэглэгчид хүргэсний дараа алдааг засах, гүйцэтгэл болон бусад үзүүлэлтийг сайжруулах, орчинд тохируулахын тулд програм хангамжийн бүтээгдэхүүнийг өөрчлөх үйл ажиллагаа юм [2]. Иймээс алдаагүй зөв програм хангамж хөгжүүлсэнээр програм хангамж хөгжүүлэлтийн зардлыг бууруулах боломжтой. Мөн түүнчлэн засвар үйлчилгээ хийх зардлын багагүй хэсэг нь програмын үйл ажиллагаа, бүтцийг ойлгох, түүнийг хэрхэн засварлахыг тодорхойлоход зарцуулагддаг. Тиймээс програм хангамжийг алдаагүй зөв бүтэцтэй хөгжүүлэх шаардлагатай.

Програм хангамж хөгжүүлэлтийн эрт үед гарсан алдаа нь дараа дараачийн үе шатуудад алдаа гарах суурь болдог бөгөөд үүнээс үүдэн алдааны тоо алхам тус бүрд огцом өсдөг [3]. Тиймээс програм хангамж хөгжүүлж дууссаны дараа алдааг илрүүлэх бус харин хөгжүүлж байх явцад алдаагүй програм хангамж хөгжүүлэхэд анхаарах хэрэгтэй.

Объект хандлагат програм хангамж хөгжүүлэх үед үндсэн бүтэц болох классын алдаанд өртөмтгий байдлыг зохиомжын үе шатанд илрүүлэх аргыг тодорхойлсоноор алдаагүй зөв програм хангамж хөгжүүлэх боломжийг нэмэгдүүлнэ.

Програм хангамжийн чанарыг эрт үед илрүүлэх нэг арга нь зохиомжийн хэмжигдэхүүн ашиглах юм. Мөн өмнө үүссэн мэдээлээс суралцан өгөгдсөн мэдээлэл бүхий өгөгдөл аль ангилалд хамаарахыг таамаглах нэг арга нь ангилах алгоритм ашиглах юм.

B. Объект хандлагат зохиомжийн хэмжигдэхүүн

Энэхүү хэсэгт түгээмэл ашиглагддаг объект хандлагат зохиомжийн хэмжигдэхүүнүүдийг танилцууллаа.

MOOSE metrics [4] – Чидамбер болон Кемерер нар нь нар нь объект хандлагат зохиомж боловсруулах эхний гурван алхамын үр дүнг хэмжих Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Object classes (CBO), Response For a Class (RFC) болон Lack of Cohesion in Methods (LCOM) хэмжигдэхүүнийг тодорхойлсон

EMOOSE [5] metrics – Ли нь MOOSE хэмжигдэхүүнийг Message Pass Coupling (MPC), Data Abstraction Coupling (DAC), Number of Methods (NOM), Size1 болон Size2 гэсэн хэмжигдэхүүнээр өргөтгөн тодорхойлсон.

MOOD Metrics [6] - Абрей нь бүтцийн загварын объект хандалтад үндсэн шинжүүдийг тодорхойлох битүүмжлэлийн Method Hiding Factor (MHF), Attribute Hiding Factor (AHF), удамшлийг илэрхийлэх Method Inheritance Factor (MIF), Attribute Inheritance Factor (AIF), полиморфизмийг илэрхийлэх Polymorphism Factor (POF), болон зурвас дамжуулалтыг илэрхийлэх Coupling Factor (COF) хэмжигдэхүүнийг санал болгосон.

MOOD2 [6] Metrics – MOOD хэмжигдэгдэхүүнийг Operation Hiding Effectiveness Factor (OHEF), Attribute

Hiding Effectiveness Factor (AHEF), Internal inheritance factor (IIF) болон Parametric polymorphism factor (PPF) гэсэн хэмжигдэхүүнүүдээр өргөтгөсөн.

QMOOD metrics [5] - Бансияа нь объект хандлагат зохимжийн дахин ашиглалт, уян хатан байдал, өргөтгөж болохуйц байдал, үр дүнтэй байдал болон ойлгомжтой байдлыг тодорхойлох зорилгоор үүсгэсэн. Эдгээр хэмжигдэхүүнүүд нь Design Size in classes (DSC), Number of Hierarchies (NOH), Average Number of Ancestors (ANA), Data Access Metric (DAM), Direct Class Coupling (DCC), Class interface Size (CIS), Measure of aggregation (MOA), Cohesion Among Methods of Class (CAM), Measure of Functional Abstraction (MFA), Number of Polymorphic methods (NOP), Number of methods (NOM).

Lorenz and Kidd Metrics [7] – Лоренз болон Кейд нар нь зохиомжийн үндсэн гурван үзүүлэлт болох хэмжээ (Number of Public Methods (NPM), Number of Methods (NM), Number of Public Variables per class (NPV), Number of Variables per class (NV), Number of Class Variables (NCV), Number of Class Methods (NCM)), удамшил (Number of Methods Inherited (NMI), Number of Methods Overridden (NMO), Number of New Methods (NNA)) классын нягтралыг (Average parameters per Method (APM), Specialization Index (SIX)) тодорхойлох хэмжигдэхүүнүүдийг санал болгосон.

II. Холбоотой ажлын судалгаа

Объект хандлагат зохиомж (ОХЗ)-ийн хэмжигдэхүүн болон классын алдаанд өртөмтгий байдал хоорондын хамаарлыг илрүүлэх судалгааг цөөнгүй хийсэн байдаг бөгөөд [8,9,10] судалгааны ажлууд програм хангамж хөгжүүлэлтийн тестийн үе шатанд алдаа гарсан байх магадлалтай бүтцийг илрүүлэх зорилгоор хийгдсэн. Харин [11,12,13,14,15,16,17,18,19,20] судалгааны ажлууд програм хангамж хөгжүүлэлтийн хэрэгжүүлэлтийн үед классын алдаанд өртөмтгий байдал болон хэмжигдэхүүн хоорондын хамаарлыг тодорхойлсон. Эдгээр судалгааны ажлууд нь бүгд EMOOSE хэмжигдэхүүнийг бусад зохиомжийн хэмжигдэхүүнтэй хослуулан ашигласан байсан.

ОХЗ-ийн хэмжигдэхүүн ашиглан классын алдаанд өртөмтгий байдлыг тодорхойлохдоо [9,15,16,21] судалгааны ажлууд регрессийн арга ашигласан бөгөөд хамгийн өндөр F-Measure үзүүлэлттэй байсан нь [15] судалгааны ажил байсан. Харин [9,22,23] нь тус бүр машин сургалтын хэд хэдэн аргыг ашиглан классын алдаанд өртөмтгий байдлыг таамаглахад хамгийн зохимжит алгоритмыг тодорхойлохыг зорьсон. Эдгээр туршилтын үр дүнд хамгийн өндөр үр дүнтэй байсан алгоритм нь Random Forest байсан.

III. ХЭРЭГЖҮҮЛЭЛТ

A. Хэмжигдэхүүн сонголт

Энэхүү судалгааны ажилд ашиглагдах хэмжигдэхүүн нь класс диаграмаас тооцоолох боломжтой, нэгэн ижил үзүүлэлт хэмждэггүй байх бөгөөд объект хандлагатын үндсэн шинжүүдийг тодорхойлогч байх шаардлагатай.

Эдгээр шаардлагыг хангасан хэмжигдэхүүн сонгохын тулд дараах алхамуудыг дэс дараалан гүйцэтгэсэн.

- Судалгааны үед цуглуулсан нийт зохиомжийн хэмжигдэхүүнээс систем болон багцын үзүүлэлтийг илэрхийлэх хэмжигдэхүүнийг хасах.
- Өмнөх алхмын үр дүнд үлдсэн хэмжигдэхүүнээс классын зохиомжийн загвараас тооцоолоход хүндрэлтэй, тооцоолох боломжгүй хэмжигдэхүүнийг хасах.
- Ижил төстэй үзүүлэлтийг хэмждэг хэмжигдэхүүнийг тодорхойлох, давхцсан хэмжигдэхүүнийг хасах.
- Сонгогдсон хэмжигдэхүүн нь объект хандалтад зохиомжийн үндсэн шинж тус бүрийн үзүүлэлтийг тодорхойлж чадаж байгааг шалгах.

Дээрх алхамуудын үр дүнд ХҮСНЭГТ 1-д тодорхойлсон хэмжигдэхүүнийг ашиглахаар болсон.

ХҮСНЭГТ 1. ОБЪЕКТ ХАНДАЛАГТ ЗОХИОМЖИЙН ХЭМЖИГДЭХҮҮН

Зохиомжийн шинж	Хэмжигдэхүүн
Цогц байдал	SIZE2, NPA, NOA, NCA, NCM, NNA, WMC
Өгөгдөл нууцлал	MHF, AHF, DAM, OAM, MAM, CIS,
Удамшил	DIT, NOC, MIF, AIF, MAT, MFA, NMI,
Дахин ашиглалт	POF, NMO, SIX
Холболт	CBO, CFO, MFM, MOA, MRM, COF
Нягтшил	RFC, MOS, CAM, APM

B. Туршилтын өгөгдөл бэлтгэх

Энэхүү судалгааны ажилд системд гарсан бодит алдааны бүртгэлийн мэдээлэл хэрэгтэй бөгөөд Мензис болон түүний багийн цуглуулсан нээлттэй эхийн системүүд дээр гарсан алдааны бүртгэл Promise Data санг [24] ашигласан бөгөөд ашигласан төслүүд болон түүнд гарсан алдааны мэдээллийг ХҮСНЭГТ 2-д харууллаа.

ХҮСНЭГТ 2. Туршилтын өгөгдөл

Төсөл	Нийт классын тоо	Нийт алдааны тоо	Алдаатай классын тоо
Ant1.5	270	35	32
Ant1.6	325	172	85
Ant1.7	720	312	152
jEdit3.2	268	382	90
Huim	1583	901	359

Дээрх төслүүдийн эх кодоос AltovaUML багажийн тусламжтай классын диаграмыг боловсруулсан бөгөөд класс диаграмыг ХМI бүтэцээр хадгалсан. Зохиомжийн үзүүлэлт хэмжихэд шаардагдах зарим дүрмийг нэмж тодорхойлсон UML22Measure програмыг ашиглан .xmi өргөтгөлтэй файлаас зохиомжийн үзүүлэлтийг хэмжсэн. Хэмжилтийн үр дүнг класс тус бүрээр алдааны мэдээлэлтэй холбож судалгаанд ашиглах өгөгдөл бэлтгэсэн.

С. Зохиомжийн хэмжигдэхүүн болон классын алдаанд өртөмтгий байдлын хамаарлыг тодорхойлох

Класс диаграмаас хэмжсэн зохиомжийн үзүүлэлт болон классын алдаанд өртөмтгий байдал хоорондын хамаарлыг тодорхойлох зорилгоор хэмжигдэхүүн тус бүр болон алдаанд өртөмтгий байдал хоорондын Pearson-ийн хамаарлын коэффициентийг тооцоолсон. Тодорхойлогч утгыг тооцоолохдоо хоёр талт тестийн арга ашигласан.

Хамаарлын шинжилгээний үр дүнд нийт 33 хэмжигдэхүүнээс 25 хэмжигдэхүүн нь классын алдаанд өртөмтгий байдлыг 95%-ийн магадлалтай тодорхойлогч үзүүлэлт болж байна. Эдгээрт объект хандалтадын шинж тус бүрийн хэмжигдэхүүн орсон. Энэ нь классын алдаанд өртөмтгий байдал нь объект хандалтадын бүх шинжээс хамаардаг болохыг харуулж байна. Мөн түүнчлэн класс диаграмаас тухайн классын алдаанд өртөмтгий байдлыг таамаглах боломжтойг харуулж байна.

D. Ангилах алгоритм сонголт

Классын алдаанд өртөмтгий байдлыг таамаглахдаа түгээмэл ашиглагддаг ангилах аргуудаас Naive Bayes, Multilayer Perception, Instance Based, Classification Via Regression, Decision Table болон Random Forest алгоритм ашигласан.

Эдгээр алгоритмаас аль нь классын алдаанд өртөмтгий байдлыг таамаглахад илүү тохиромжтой болохыг тодорхойлох зорилгоор туршилтыг дараах дөрвөн тохиолдолд явуулсан. Үр дүнг accuracy, precision, Recall (алдаатай классыг зөв таамагласан) болон F-Measure үзүүлэлтээр харьцуулсан.

Туршилт 1. Урьдчилан боловсруулаагүй өгөгдлөөс алдааг илрүүлэх

Туршилт 2. Шинж сонгосон өгөгдлөөс алдааг илрүүлэх

Туршилт 3. Өгөгдлийн гажгийг зассан өгөгдлөөс алдааг илрүүлэх

Туршилт 4. Өгөгдлийн гажгийг засаж, шинж сонгосон өгөгдлөөс алдааг илрүүлэх

Шинж сонгох Correlation based subset selection болон өгөгдлийн гажгийг засах Resampling алгоритмыг ашигласан бөгөөд Weka багажийг ашиглан туршилтыг гүйцэтгэсэн.

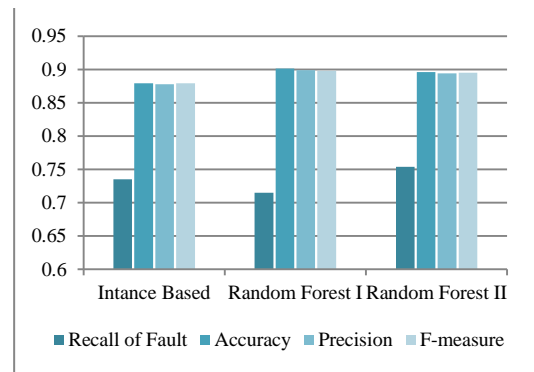
Туршилт тус бүрд хамгийн өндөр үр дүнтэй байсан алгоритмыг ХҮСНЭГТ 3-д харууллаа.

ХҮСНЭГТ 3. Туршилтын үр дүн

	Алгоритм	Үр дүн			
		Accuracy	Precision	Recall of fault	F-measure
1	Naive Bayes	76.394	0.783	0.568	0.772
	Classification Via Regression	81.784	0.80	0.373	0.798
2	Naive Bayes	81.227	0.795	0.415	0.799
	Classification Via Regression	81.97	0.81	0.263	0.784
3	Instance Based	87.918	0.878	0.738	0.879
	Random Forest	89.405	0.892	0.662	0.889
4	Random Forest	87.732	0.873	0.662	0.873

Туршилтын үр дүнгээс харвал хамгийн өндөр танилтын хувьтай алгоритм нь Random Forest, алдаатай классыг хамгийн сайн таамаглаж байсан нь Instance Based алгоритм байна. Random Forest нь сонгосон модны тооноос хамааран өөр өөр үр дүн өгдөг учир тохиромжтой модны тоо сонгох туршилтыг 3 болон 4-р тохиолдолд дахин туршихад харгалзан 20 болон 15 мод сонгоход хамгийн өндөр үр дүнтэй байсан.

Нийт туршилтын хувьд хамгийн өндөр үр дүнтэй алгоритмуудын харьцуулалтыг Зураг 1-д харуулсан. Тус үр дүнгээс харахад өгөгдлийн гажгийг засаж дэд олонлог сонгосон өгөгдлөөс классын алдаанд өртөмтгий байдлыг Random Forest алгоритмаар 15-н санамсаргүй мод үүсгэн таамаглах нь хамгийн өндөр үр дүнтэй арга байна. Тус арга нь нийт алдаатай классын 75%-ийг зөв таамаглан ойролцоогоор 90%-ийн танилтын үзүүлэлттэй байна.



Зураг 1. Туршилтын үр дүн

IV. ҮР ДҮН, ХАРЬЦУУЛАЛТ

A. Хамаарлын шинжилгээний үр дүн

Объект хандалтад зохиомжийн хэмжигдэхүүн болон классын алдаанд өртөмтгий байдал хоорондын хамаарлын чиглэлд үндэслэн классын зохиомжийн загвар гаргахад анхаарвал зохих зөвлөмж гаргасан.

- Аргыг аль болох бага дахин тодорхойлох.
- Ерөнхийллийг өндөр түвшинд сайн ашиглах.
- Класс тус бүрд агуулагдах мэдээлэл болон боловсруулах үйлдлийн тоо аль болох цөөн байх.
- Нэг класс аль болох цөөн класстай хамтарч ажилладаг байх.
- Шаардлагагүй тохиолдолд өгөгдөл нууцлалыг аль болох бага хэрэглэх.

Хамаарлын шинжилгээний үр дүнд тодорхойлсон тодорхойлогч утгуудыг бусад судлаачдын хийсэн хамаарлын шинжилгээний үр дүнтэй харьцуулж үзсэн. Тус чиглэлээр гүйцэтгэсэн судалгааны ажлын цөөнгүй хэсэг нь MOOSE хэмжигдэхүүнийг ашигласан байсан бөгөөд тэдгээр судлаачдын үр дүнг өөрийн үр дүнтэй харьцуулсан үзүүлэлтийг ХҮСНЭГТ 4-д харууллаа.

ХҮСНЭГТ 4. ХАМААРЛЫН ШИНЖИЛГЭЭНИЙ ҮР ДҮНГИЙН ХАРЬЦУУЛАЛТ

Зохиогч	WMC	DIT	RFC	NOC	CBO	LCOM
Basili et al.	+	++	++		+	0
Tang Briand et al.	+	0	+	0	0	
Briand et al.	++	++	++	-	++	
Briand et al.	+	--	++	0	++	
Emam et al.	+	0	++		+	
Yu et al.	++	0	+	++	+	
Subramanyam et al.	+	-			+	
	0	-			-	
Gyimothy et al.	++	+	++	0	++	+
Zhou et al.	++	0	++	--	++	+
Olague et al.	++	0	++	0	++	++
Sing et al.	++	0	++	0	++	+
Судалгааны үр дүн	++	0	++	0	++	

Энд: + нь +0.05, - нь -0.05, ++ нь +0.01, -- нь -0.01 –ийн тодорхойлогч үзүүлэлт болохыг илэрхийлэх бол 0 нь тодорхойлогч болж чадахгүйг, хоосон нүд нь тухайн шинжилгээнд ашиглагдаагүй болохыг илэрхийлнэ.

ХҮСНЭГТ 4-өөс харахад хамаарлын шинжилгээний үр дүн бусад судлаачдын үр дүнтэй төстэй байгаа нь зохиомжийн загварын үзүүлэлт нь кодын зохиомжийн үзүүлэлтийн нэгэн адил классын алдаанд өртөмтгий байдлыг тодорхойлогч үзүүлэлт болж чадах нь харагдаж байна. Энэ нь зохиомжийн загвараас ОХЗ-ийн хэмжигдэхүүн ашиглан классын алдаанд өртөмтгий байдлыг тодорхойлох боломжтойг илэрхийлж байна.

В. Алдааг таамаглах туршилтын үр дүн

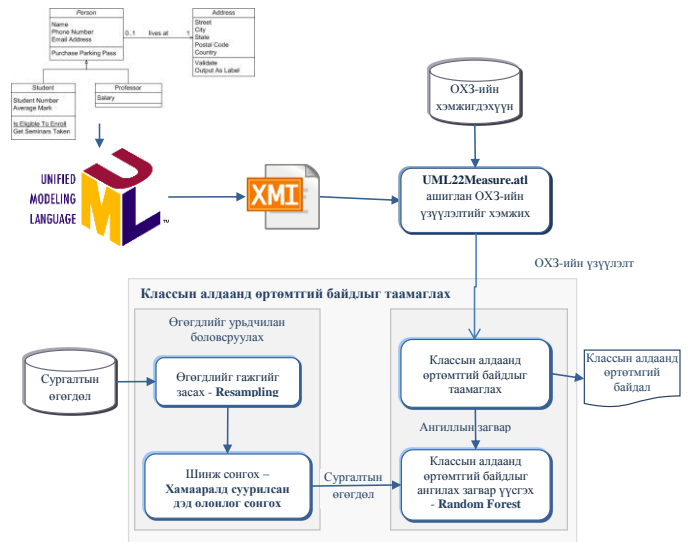
Алдааг таамаглах туршилтын үр дүн бусад судлаачдын үр дүнтэй нэгтгэн харьцуулаж ХҮСНЭГТ 5-д харуулсан.

ХҮСНЭГТ 5. ТААМАГЛАХ ТУРШИЛТЫН ҮР ДҮНГИЙН ХАРЬЦУУЛАЛТ

Зохиогч	Үе шат	Аргачлал	Accuracy	F-Measure
Fioravanti et al.	Туршилт	MLR	84.96	78.33
Briand et al.	Туршилт	MLR	79.64	92.45
Aggarwal et al.	Туршилт	MLR	93.75	81.89
Surbhi Gaur et al.	Туршилт	Logistic	81.25	90.75
Ruchika Malhotra	Хэрэгжүүлэлт	Random forest	78.6	80.7
Туршилтын үр дүн	Зохиомж	Random forest I	90.15	89.8
		Random forest II	89.6	89.5

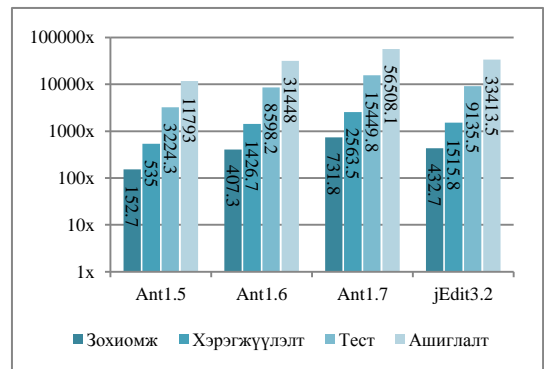
С. Санал болгож буй арга, түүний ач холбогдол

Классын зохиомжоос классын алдаанд өртөмтгий байдлыг таамаглахдаа Зураг 2-д үзүүлсэн загварыг ашиглах боломжтой. Тус загвар нь класс диаграмаас тусгай UML багаж ашиглан диаграмыг ХМI бүтцэд хөрвүүлнэ. Уг файлаас объект хандалтад зохиомжийн үзүүлэлтийг хэмжинэ. Нөгөө талаас урьдчилан бэлдсэн сургалтын өгөгдлийг ашиглан классын алдаанд өртөмтгий байдлын загвар үүсгэх бөгөөд тус загварыг ашиглан зохиомжийн загварын класс тус бүрийн алдаанд өртөмтгий байдлыг тодорхойлно.



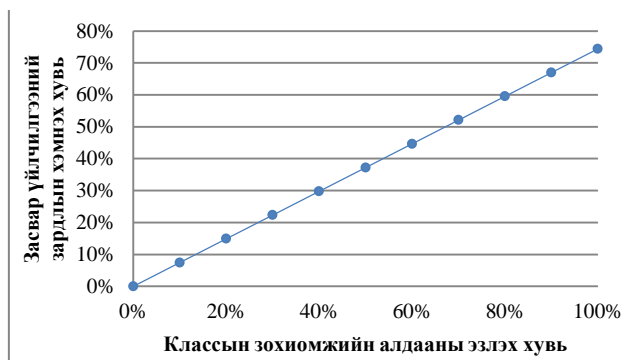
Зураг 2. Алдаанд өртөмтгий байдлыг таамаглах загвар

Програм хангамжид гарсан алдааг програм хангамж хөгжүүлэлтийн үе шат тус бүрт нэг алдааг засах өртөг ямар байхыг тооцоолсон судалгааны ажлууд цөөнгүй байдаг [25,26,27] бөгөөд тэдгээр судалгааны ажлын дундаж үр дүнг ашиглан туршилтад ашигласан төсөл тус бүрийн алдааг илрүүлэн засахад ямар өртөг зарцуулахыг систем хөгжүүлэлтийн үе шат тус бүрээр тооцоолсон. Тус тооцооллод нийт гарсан алдааг 100% зохиомжийн бүтцээс үүдэлтэй алдаа гэж үзсэн. Зохиомжийн загвараас классын алдаанд өртөмтгий байдлыг таамаглах загвар нь нийт алдаатай классын 75.4%-ийг зөв илрүүлнэ гэж үзсэн. Үр дүнг Зураг 3-д харууллаа.



Зураг 3. Туршилтад ашигласан төслүүдийн алдааг засварлахад зарцуулах өртөг

Зохиомжийн үе шатанд тус аргачлалыг ашиглан классын зохиомжийн алдааг илрүүлснээр нийт засвар үйлчилгээний зардлын хэдэн хувийг хэмнэх боломжтойг Зураг 4-д харуулсан.



Зураг 4. Засвар үйлчилгээний зардлын хэмнэлт

V. ДҮГНЭЛТ

Энэхүү судалгааны ажлын шинэлэг тал нь классын алдаанд өртөмтгий байдлыг програм хангамж хөгжүүлэлтийн зохиомжийн үе шатанд таамаглахыг зорьсон бөгөөд судалгааны ажлын үр дүнд зохиомжийн загвараас тухайн зохиомжийн дагуу үүсгэсэн програм хангамжийн класс нь алдаанд өртөмтгий эсэхийг таамаглах боломжтойг бусад судлаачдын үр дүнтэй харьцуулан тодорхойллоо. ОХЗ-ийн хэмжигдэхүүн болон алдаанд өртөмтгий байдлын хоорондын хамаарал дээр үндэслэн классын зохиомжийг гаргахад анхаарвал зохих зөвлөмж гаргасан.

Мөн түүнчлэн судалгааны үр дүнд үүсгэсэн систем хөгжүүлэлтийн зохиомжийн үе шатанд классын алдаанд өртөмтгий байдлыг таамаглах аргачлал нь нийт алдаатай классын 75%-г таних боломжтой бөгөөд F-Measure үзүүлэлт нь ойролцоогоор 90% байгаа нь тус загварыг бүрэн ашиглах боломжтой болохыг харуулж байна.

Тус загварыг програм хангамжийн зохиомжийн үе шатанд ашигласнаар алдаагүй зөв бүтцийг зохиомжлоход туслах бөгөөд энэ нь кодчиллын үед зөвхөн кодын бүтцэд л анхаарлаа хандуулах боломжийг олгоно. Классын алдааг эрт илрүүлж зассанаар алдааг засахад зарцуулах зардлыг бууруулах бөгөөд энэ нь програм хангамж хөгжүүлэлтийн зардлыг багасгах нэгэн хүчин зүйл болно.

НОМ ЗҮЙ

- [1] Harald H. Vogt Floris P. Engelbertink, "How to save on software maintenance," , 2010.
- [2] Computer Society, IEEE Standard for Software Maintenance, 1993.
- [3] Information Technology Promotion Agency, *System Development and Operation*, 3rd ed. Tokyo, Japan: Japan Information Technology Engineers Examination Center, 2003.
- [4] S. R. and C. F. Kemerer Chidamber, "A Metrics Suite for Object Oriented Design," no. 476-793, 1994.
- [5] C. G. Davis J. Bansiya, "A Hierarchical Model for Object-Oriented Design Quality Assessment," , IEEE Transactions on Software Engineering, 28, 2002.
- [6] Rogério Carapuça Fernando Brito e Abreu, "Object-Oriented Software Engineering: Measuring and Controlling the Development Process," in *4th Int. Conf. on Software Quality*, McLean, VA, USA, 1994, pp. 3-5.
- [7] M. Lorenz and J. Kidd, *Object-Oriented Software Metrics*. New Jersey, Englewood Cliffs: Prentice Hall, 1994.

- [8] K., W.Melo, and J.Machado. El Emam, "The prediction of faulty classes using object-oriented design metrics," *The Journal of Systems and Software*, no. 56(1), pp. 63-75, 2001.
- [9] T., Ferenc, R., & Siket, I Gyimothy, "Empirical validation of object-oriented metrics on open source software for fault prediction," *IEEE Transactions on Software Engineering*, vol. 29, pp. 297-310, 2003.
- [10] Y., & Leung, H. Zhou, "Empirical analysis of object oriented design metrics for predicting high severity faults," *IEEE Transactions on Software Engineering*, no. 32(10), pp. 771-784, 2006.
- [11] V., Briand, L., & Melo, W. Basili, "A validation of object oriented design metrics as quality indicators," *IEEE Transactions on Software Engineering*, no. 22(10), pp. 751-761, 1996.
- [12] M. H., Kao, M. H., & Chen, M. H. Tang, "An empirical study on object-oriented metrics," in *Proceedings of 6th IEEE International Symposium on Software Metrics*, pp. 242-249, 1999.
- [13] K., Benlarbi, S., Goel, N., & Rai, S. El Emam, "The confounding effect of class size on the validity of object oriented metrics," *IEEE Transactions on Software Engineering*, no. 27(7), pp. 630-650, 1999.
- [14] K., Benlarbi, S., Goel, N., & Rai, S. El Emam, "A validation of object-oriented metrics," NRC, Technical report ERB-1063 1999.
- [15] Jürgen Wüst, John W. Daly, and D. Victor Porter, Lionel C. Briand, "Exploring the Relationships between Design Measures and Software Quality in Object-Oriented Systems," vol. 51, no. 3, 2000.
- [16] F., & Nesi, P. Fioravanti, "A Study on Fault-Proneness Detection of Object-Oriented Systems," in *Fifth European Conference Software Maintenance and Reengineering*, 2001, pp. 121-130.
- [17] P., Systa, T., & Muller, H. Yu, "Predicting fault-proneness using OO metrics : An industrial case study," in *Proceedings of Sixth European Conference on Software Maintenance and Reengineering*, Budapest, Hungary, 2002, pp. 99-107.
- [18] R. and M.S. Krishnan Subramanyam, "Empirical analysis of CK metrics for object oriented design complexity. Implications for software defects," *IEEE Trans. Software Eng.*, vol. 29, pp. 297-310, 2003.
- [19] H., Etkorn, L., Gholston, S., & Quattlebaum, S Olague, "Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes," *IEEE Transactions on Software Engineering*, no. 33(8), pp. 402-419, 2007.
- [20] Y., Kaur, A., & Malhotra, R Singh, "Empirical validation of object-oriented metrics for predicting fault proneness models," *Software Quality Journal*, pp. 3-35, 2010.
- [21] Yogesh Singh, Arvinder Kaur, Ruchika Malhotra K.K. Aggarwal, "Investigating effect of Design Metrics on Fault Proneness in Object-Oriented System," vol. 6, no. 10, 2007.
- [22] "Validation of Software Quality Models using Machine Learning: An Empirical Study," *International Journal of Computer Applications*, vol. 8, 2013.
- [23] and Ankita Jain Ruchika Malhotra, "Fault Prediction Using Statistical and Machine Learning Methods for Improving Software Quality," vol. 8, 2012.
- [24] B. Caglayan, E. Kocaguneli, J. Krall, F. Peters, and B. Turhan T. Menzies, *The PROMISE Repository of empirical software engineering data.*: West Virginia University, Department of Computer Science, 2012.
- [25] Barry W. Boehm, *Software engineering economics*. Englewood Cliffs: Prentice Hall, 1981.
- [26] Jonette M. Dabney, Jim. Dick, Brandon. Haskins, Bill Stecklein, "Error Cost Escalation Through the Project Life Cycle," in *14th Annual International Symposium*, Toulouse; France, 2014.
- [27] E Wainright E Martin et al., *Managing Information Technology*, 4th ed.: Prentice Hall, 2001.
- [28] G Booch, "Object Oriented Design with Applications," 1991.