

კოდში შეცდომების გამოვლენის ავტომატიზაცია ხელოვნური ინტელექტის გამოყენებით

ირაკლი ნოზაძე

სამაგისტრო ნაშრომი წარდგენილია ილიას სახელმწიფო უნივერსიტეტის ბიზნესის,
ტექნოლოგიისა და განათლების ფაკულტეტზე მაგისტრის აკადემიური ხარისხის
მინიჭების მოთხოვნების შესაბამისად

პროგრამული ინჟინერია

ნაშრომის ხელმძღვანელი: **გიორგი ფარცხალაძე**, ილიას სახელმწიფო
უნივერსიტეტის ასოცირებული პროფესორი

პროექტის ხელმძღვანელი: **პაატა გოგიშვილი**



ილიას სახელმწიფო უნივერსიტეტი
თბილისი 2025

განაცხადი

როგორც წარდგენილი სამაგისტრო ნაშრომის ავტორი, ვაცხადებ, რომ ნაშრომი ჩემი ორიგინალური ნამუშევარია და არ შეიცავს სხვა ავტორების მიერ აქამდე გამოქვეყნებულ, გამოსაქვეყნებლად მიღებულ ან დასაცავად წარდგენილ მასალებს, რომლებიც არ არის მოხსენიებული ან ციტირებული სათანადო წესების შესაბამისად.

ირაკლი ნოზაძე

30 ივნისი 2025

მადლობა

მინდა გულწრფელი მადლობა გადავუხადო ჩემს დისერტაციის ხელმძღვანელს, კოლეგებსა და მეგობრებს კვლევის მთელი პროცესის განმავლობაში მათი ფასდაუდებელი მხარდაჭერისა და დახმარებისთვის.

ირაკლი ნოზაძე

აბსტრაქტი

პროგრამული უზრუნველყოფის სწრაფ განვითარების ფონზე, კოდის საიმედოობის უზრუნველყოფა წარმოადგენს დეველოპმენტის პროცესის მდგრადობისა და ეფექტურობის ძირითად პირობას. თანამედროვე კოდის შემუშავება მოითხოვს კოდში როგორც სინტაქსური, ისე ლოგიკური შეცდომების სწრაფი აღმოჩენის შესაძლებლობას.

ბოლო წლებში დიდი ენის მოდელების (Large Language Models) განვითარებამ მნიშვნელოვნად გაამარტივა ე.წ. დეფექტების (Defects) აღმოჩენა პროგრამულ კოდში. თუმცა, ასეთი მოდელები ყველა პროგრამირების ენისთვის ან არ არსებობს, ან მძიმეა და არ არის კონკრეტული IT ორგანიზაციების საჭიროებებზე მორგებული.

აღნიშნული სამაგისტრო ნაშრომი მიზნად ისახავს Python-ის პროგრამირების ენის მაგალითზე მსუბუქი ხელოვნური ინტელექტის მოდელის შემქნას, რომელიც მარტივად ინტეგრირდება გაფართოების (extension) სახით ტექსტურ რედაქტორებში, როგორცაა, მაგალითად, Visual Studio Code. მოდელი უნდა უზრუნველყოფდეს შეცდომების რეალურ დროში (Just-In-Time, JIT) აღმოჩენას და შედეგების მომხმარებლისთვის გაზიარებას. მთავარი უპირატესობა, რის საშუალებასაც მოდელი უნდა იძლეოდეს, არის მისი ლოკალურობა, მოდელის მცირე ზომა და მომხმარებლის სრული კონტროლი მოდელის მუშაობის პროცესზე.

ნაშრომის ფარგლებში აღებულ იქნა Transformer-ის ტიპის CodeBERTa-Small მოდელი და გაიწვრთნა Defectors-ის (A Large, Diverse Python Dataset for Defect Prediction) მონაცემთა ნაკრებზე და მიღწეულ იქნა კოდში ყოველი ხაზისთვის დეფექტის აღმოჩენის 98.2% სიზუსტე (Accuracy).

აღნიშნულ კვლევაში ასევე გამოკვლეულია, თუ როგორ შეუძლია დეფექტების პროგნოზირების სისტემამ მნიშვნელოვნად გააუმჯობესოს პროგრამული უზრუნველყოფის ინჟინრების ყოველდღიური სამუშაო პროცესი.

ძირითადი საძიებო სიტყვები: CodeBERTa-Small, შეცდომების დროული აღმოჩენა (JIT), Visual Studio Code-ის გაფართოება, ტრანსფორმატორზე დაფუძნებული კოდის ანალიზი, სემანტიკური დეფექტების პროგნოზირება.

Abstract

In the context of rapid software development, ensuring code reliability is a key prerequisite for the stability and efficiency of development processes. Modern code development requires the ability to quickly identify both syntactic and logical errors.

In recent years, the advancement of Large Language Models (LLMs) has significantly simplified the detection of so-called *Defects* in source code. However, such models are either unavailable for all programming languages or are too resource-intensive and not tailored to the specific needs of individual IT organizations.

This master's thesis aims to develop a lightweight artificial intelligence model for the Python programming language that can be easily integrated as an extension into text editors such as Visual Studio Code. The model is expected to support Just-In-Time (JIT) error detection and provide feedback to the user in real time. Its key advantages include local deployment, compact model size, and full user control over the model's behavior during operation.

Within the scope of this research, a Transformer-based model – CodeBERTa-Small – was selected and fine-tuned on the Defectors dataset (A Large, Diverse Python Dataset for Defect Prediction). The model achieved an accuracy of 98.2% in detecting defects on a per-line basis within Python source code.

This study also explores how defect prediction systems can substantially improve the daily workflows of software engineers by enabling smarter, more responsive development tools.

Keywords: CodeBERTa-Small, Just-In-Time (JIT) error detection, Visual Studio Code extension, Transformer-based code analysis, semantic defect prediction.