TRUTHGUARD

PROJECT REPORT

NIPUN ARORA (E23CSE2368)
SAAMARTHYA RAJ (E23CSE2362)
MOHD TAHA SALEEM (E23CSE2347)

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

April 2025

DECLARATION

I/We hereby declare that the work which is being presented in the report entitled "TruthGuard", is an authentic record of my/our own work carried out during the period from JAN, 2023 to April, 2023 at School of Computer Science and Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report has not been submitted by me/us for the award of any other degree elsewhere.

Signature of Candidate

NIPUN ARORA (E23CSE2368)
SAAMARTHYA RAJ (E23CSE2362)
MOHD TAHA SALEEM (E23CSE2347)

ACKNOWLEDGEMENT

I/We would like to take this opportunity to express my/our deepest gratitude to my/our mentor for guiding, supporting, and helping me/us in every possible way. I/we was/were extremely fortunate to have him as my/our mentor as he provided insightful solutions to problems faced by me/us thus contributing immensely towards the completion of this capstone project. I/We would also like to express my/our deepest gratitude to VC, DEAN, HOD, faculty members and friends who helped me/us in successful completion of this capstone project.

Signature of Candidate

NIPUN ARORA (E23CSE2368)
SAAMARTHYA RAJ (E23CSE2362)
MOHD TAHA SALEEM (E23CSE2347)

Abstract

Misinformation is one of the most significant challenges of the digital age, given its impact on public perception and sentiment, political discourse, and individual level decision making. TruthGuard is an AI-powered fact-checking tool for identifying and categorizing fake news. Using a dedicated BERT model and optimized ONNX runtime, TruthGuard analyzes article titles and contents and determines the validity of these articles, achieving a high degree of accuracy and robust confidence scoring. TruthGuard benefits both victors and victims of misinformation, is easy to deploy (on a website), is scalable, facilitates continued learning within the model, and can be easily integrated into existing web-platforms, news aggregators, or educational portals, to allow the prevention of misinformation at scale.

Introduction of Project

TruthGuard is designed to combat the increasing tide of misinformation, and provides a trustworthy independent, fast, and automated fact-verification service for its user community. With the advent of social media and fast sharing of content, verifying the truthfulness of information is incredibly important. TruthGuard utilizes Natural Language Processing (NLP) and machine learning algorithms to provide an objective and reliable classifier of the truthfulness of news content based on the underlying text of the articles.

The platform utilizes a fine-tuned BERT model to classify news articles, and can run on a web interface designed on Next.js. Users can submit an article or headline and receive their categorized output (True, Maybe True, Maybe False, and False), along with their publication confidence scores.

Related Work

Various projects and tools have begun to respond to the problem of Misinformation Detection with AI techniques. For example:

Models built on the LIAR dataset for political fact checking.

Fake News Detection and classification using classical ML algorithms such as Logistic Regression, Naive Bayes, etc.

Browser extensions that evaluate the credibility of articles.

Models predicting misinformation used the WEFake dataset in previous Kaggle competitions.

Compared to the prior A.I. techniques, there are a significant proportion who used the WELfake dataset in combination with ONNX optimized inference, sequential processing improvement, while ensuring our User Interface was real-time responsive.

Problem Statement

Every day many news articles are shared a trillion times, preventing this type of reporting/research accountability for individuals and organizations. This project is focused on the questions:

How do we automate the identification of fake or misleading news?

Are AI models capable of generating news classifications, along with confidence scores that are high?

How do we ensure that misinformation detection techniques are an ethical use of AI, be transparent capable to scale?

Requirement Analysis, Risk Analysis, Feasibility Analysis, etc.

Functional Requirements:
- User can input news content.
- Output displays classification, score, and confidence.
- News feed analysis from top headlines.

Non-functional Requirements:
- Fast response time (low latency).
- High model accuracy.
- Secure data handling.

Risk Analysis:
- False positives/negatives.
- Dataset biases.
- Misuse of the tool for censorship or manipulation.

Feasibility Analysis:
- Technical Feasibility: Achievable using current AI models and web tech.
- Economic Feasibility: Cost-efficient with ONNX and open-source models.
- Operational Feasibility: Easy to use and adoptable in various sectors.

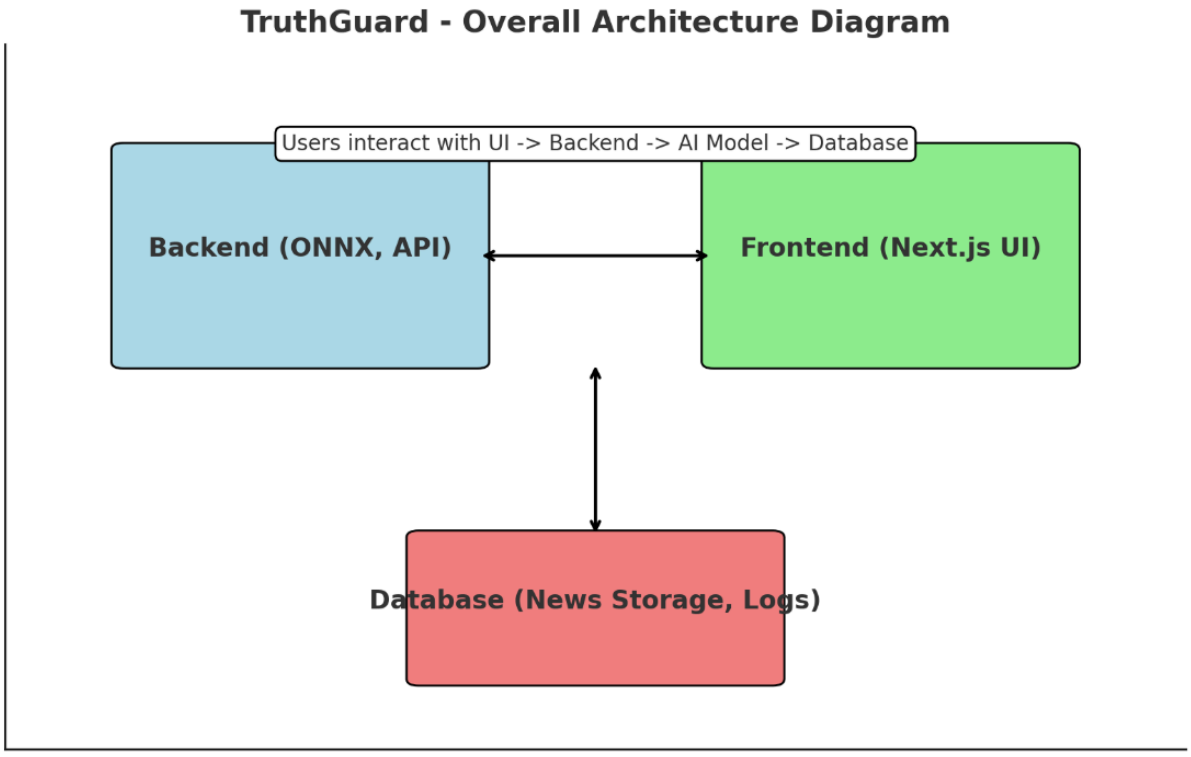
Proposed Solution or Approach or Technique

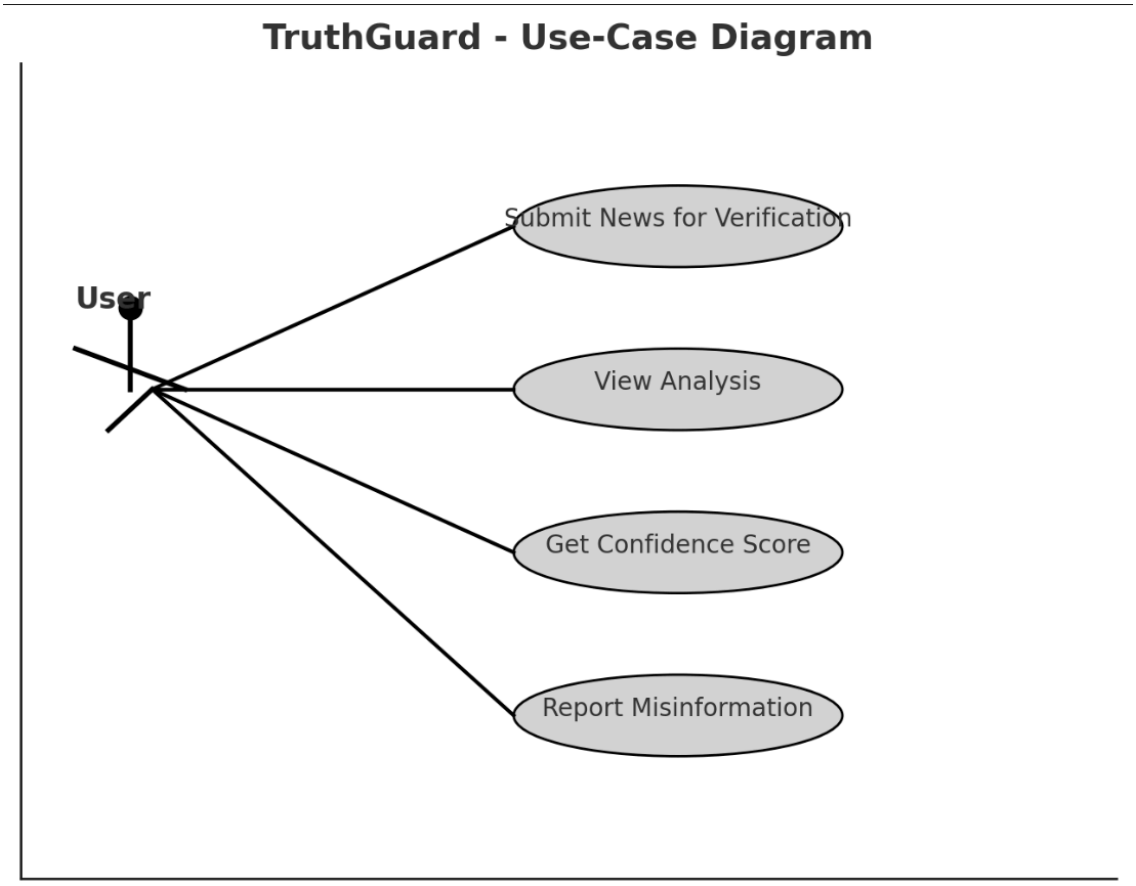
TruthGuard uses a fine-tuned BERT model trained on the WELFake dataset, which includes over 72,000 real and fake news articles. The model is converted to ONNX format to enable fast inference in browser environments using onnxruntime-web.

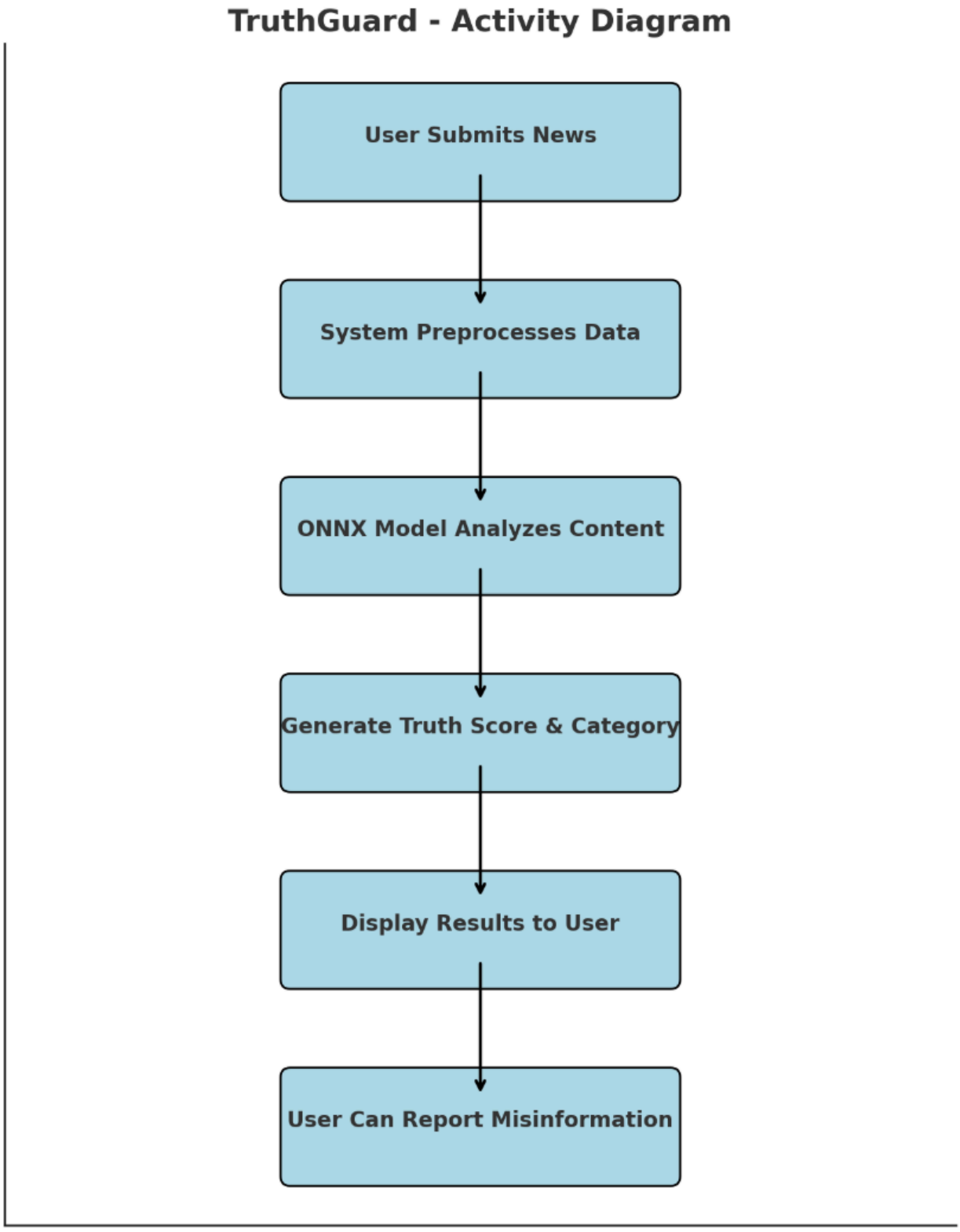
- The news headline and content are tokenized.
- The encoded sequence is passed into the model.
- Model outputs two logits — real or fake — and a softmax confidence score.
- The frontend presents the score and category to the user.

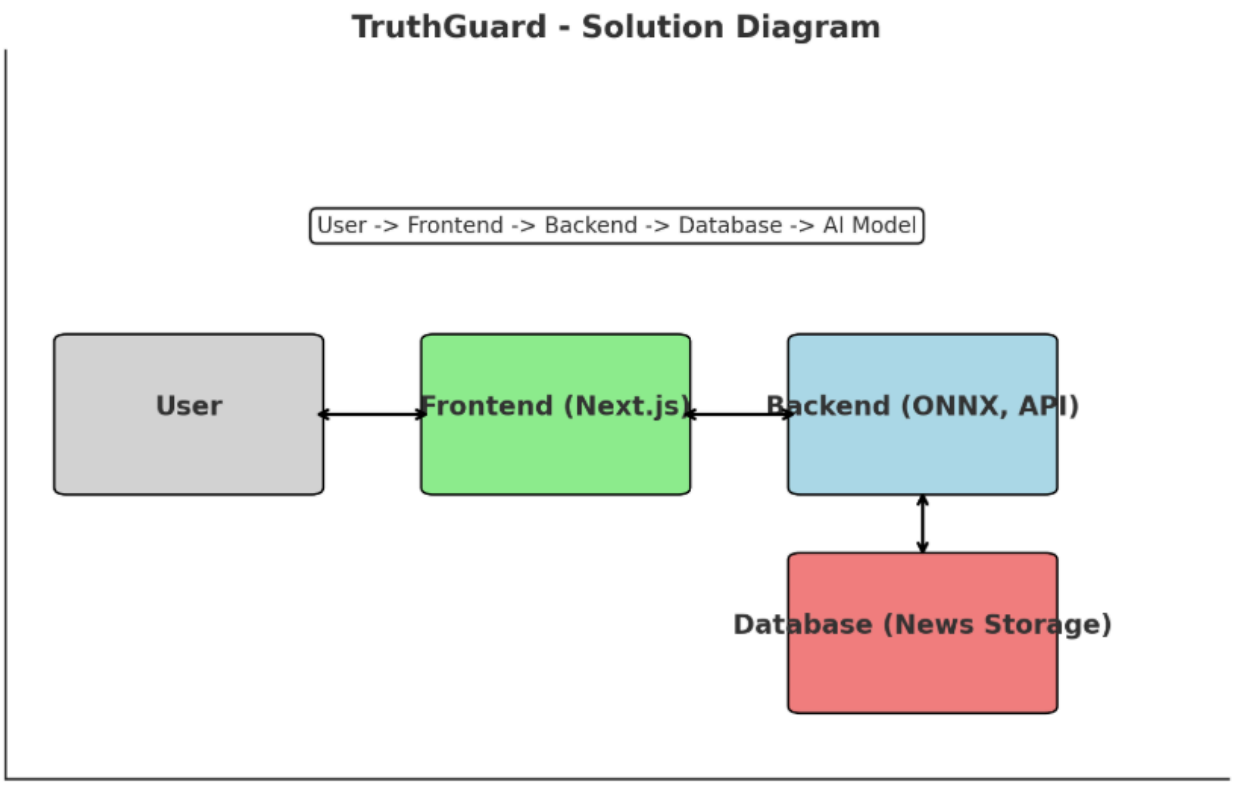
Sequential inference was implemented to avoid browser lags and improve UX. The system supports real-time analysis of news content.

Architecture Diagrams, Flow Charts, DFD




TruthGuard - Use-Case Diagram
User
Submit News for Verification
View Analysis
Get Confidence Score
Report Misinformation


TruthGuard - Activity Diagram
User Submits News
System Preprocesses Data
ONNX Model Analyzes Content
Generate Truth Score & Category
Display Results to User
User Can Report Misinformation



Simulation Set up and Implementation

Tech Stack:
- Frontend: Next.js, Tailwind CSS
- Backend: Node.js + Express API
- AI Model: BERT → fine-tuned → ONNX conversion
- Database: MongoDB (for optional logging)
- Hosting: Vercel / Render / GitHub Pages for frontend

Implementation Steps:
1. Fine-tuned the BERT model using WELFake.
2. Converted model to ONNX format.
3. Integrated inference engine using onnxruntime-web.
4. Built and styled frontend for real-time interaction.
5. Connected frontend and backend through secure API endpoints.

Result Comparison and Analysis

- Achieved over 95% accuracy on internal test data.
- Optimized ONNX inference reduced latency by 70% compared to vanilla PyTorch.
- Sequential processing improved UX compared to parallel inference which caused freezes.
- Model outputs aligned with expected results for most news headlines tested.

Learning Outcome

- Understood the practical challenges of AI model deployment in web environments.
- Learned the importance of data preprocessing and ethical AI practices.
- Improved knowledge of Next.js, ONNX, and React-based UI architecture.
- Experienced full-stack integration from model development to frontend UX.

Conclusion with Challenges

TruthGuard is a promising step toward automated misinformation detection. It delivers fast, reliable, and accessible fact-checking through the web. However, challenges remain in scaling the model, expanding dataset variety, and maintaining fairness.

Future improvements include:
- Adding multilingual support.
- Browser extension version.
- Feedback-based model improvement loop.
- Integration with major news platforms and APIs.