SVD-GRU: Robust Software Vulnerability Detection using Bayesian Gated Recurrent Unit

Orune Aminul, Dimah Dera
Department of Electrical and Computer Engineering, University of Texas Rio Grande Valley

INTRODUCTION

Software systems are prone to code defects or vulnerabilities, resulting in several problems such as deadlock, hacking, information leakage, and system failure. This research aims to develop a robust software **vulnerability detection framework** using a Bayesian gated recurrent unit (SVD-GRU) that simultaneously **predicts vulnerability** in source code and **quantifies uncertainty** in the prediction.

Table I: Statistics of the five different types of Common Weakness Enumeration (CWE) vulnerabilities

Vulnerable Class	Associated Flaws
CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer
CWE-120	Classic Buffer Overflow
CWE-469	Use of Pointer Subtraction to Determine Size
CWE-476	NULL Pointer Dereference
CWE-other	Buffer Access with Incorrect Length Value, Use of Uninitialized Variable, Improper Input Validation

BACKGROUND

Traditional Deep neural networks (DNNs) are unreliable and lack uncertainty quantification (or model confidence), which is crucial in high-stake applications, including healthcare, economy, and cyberinfrastructures [1], [2].

Our contributions:

- ❑ **Quantify uncertainty in** network's parameters and predictions.
- ❑ Develop a **robust machine learning framework**
- ❑ Learn **the mean and variance** of the predictive distribution, where the mean detects the vulnerability, and the covariance reflects the uncertainty in the predicted decision.
- ❑ **Compare** with the state-of-the-art methods in the literature and evaluate the robustness of the proposed model.

TACC ALLOCATION

This research is a part of the project '**TRUST-TRustworthy Uncertainty Propagation for Sequential Time-Series Analysis**' with TACC allocation of 20000 Sus on the Lonestar6 Supercomputer.
We have been able to make significant progress in the model evaluation within a short period of time by executing multiple jobs on the Queue (mostly normal and gpu-a100).

MATERIALS AND METHODS

Data Preprocessing

Over 1M source codes with 5 different types of CWE vulnerabilities (CWE-119, CWE-120, CWE-469, CWE-476,CWE-others) [3].

Similar to Natural Language Processing (NLP), the pre-processing steps of software source codes include **Tokenization** and word-to-vector **Embedding**

- Code is parsed to **extract tokens** of sequence length ***L***
- Converted to **vector representation**
- Embedded to further obtain into $L \times K$ representation

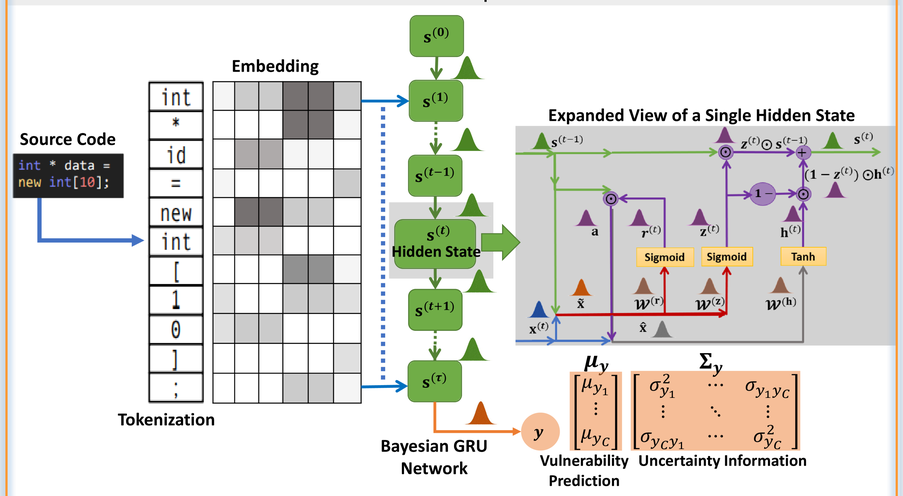

Fig 1: Illustration of the proposed software vulnerability detection approach based on Bayesian gated recurrent unit. (a) The input source code is tokenized into a token sequence of variable length τ and embedded into the τ × K matrix representation. (b) The Bayesian GRU model extracts features of the input source code from the embedding matrix and processes these features through the propagation of the variational moments. (c) The internal structure of a single GRU hidden state passes important information from the data and eliminates irrelevant ones. (d) An expanded view of the reset gate shows the interconnections between the input, x(t), hidden state, s(t−1), and reset gate output, r(t), variables. (e) The output fully connected layer classifies extracted features to detect the class vulnerability and provides the uncertainty associated with the prediction through the covariance matrix.

Bayesian GRU

- We develop a robust software vulnerability detection framework using a Bayesian gated recurrent unit (SVD-GRU).
- The proposed SVD-GRU detects source code vulnerability and simultaneously learns uncertainty in the output predictions.
- SVD-GRU adopts variational inference and optimizes the variational posterior distribution defined over the model's parameters.
- At the output of the SVD-GRU, the mean of the predictive distribution represents the vulnerability class in source codes, and the covariance matrix captures the uncertainty.

RESULTS AND DISCUSSION

Table II: Test accuracy (in %) using SVD-GRU and Deterministic GRU for different types of vulnerabilities (CWE-119, CWE-120, CWE-469, CWE-476, CWE-other, Combined Classes and Multi Head Classes respectively) under Gaussian noise, and FGSM and BIM adversarial attacks.

		Bayesian SVD-GRU							Deterministic GRU						
Noise level		C1	C2	C3	C4	C5	Combined	Multi-head	C1	C2	C3	C4	C5	Combined	Multi-head
No Noise		98	96.16	99.75	99	97.26	93.52	98	98	96	99.5	98.9	97.23	92.4	97.59
Gaussian	0.1	98	96.16	99.75	99	97.26	93.52	98	97.94	96	99.5	98.9	97.23	92.4	97.59
	0.2	97.8	96.15	99.72	98.8	97.24	93.48	97	94.4	91.92	87	92	95	88.1	95.97
	0.3	96.5	95.11	98.65	97.5	95.22	91.42	95.7	88.47	89.99	70	87.9	92.8	84.2	93.5
FGSM	0.01	97.9	96.14	99.74	98.9	97.25	93.5	97.8	97.6	94.1	98.7	98	97.2	92.02	97
	0.05	95.5	94.12	98.65	97.4	96.13	90.48	95.5	0	6.5	0	0	0.8	0.2	9
BIM	0.01	97.9	96.12	99.75	98.9	97.26	93.51	97.5	97.9	96	99	86	97	0	97
	0.05	95.2	96.1	97.72	97.4	95.22	90.45	94	0	0.3	0	2.3	2	0	9.3

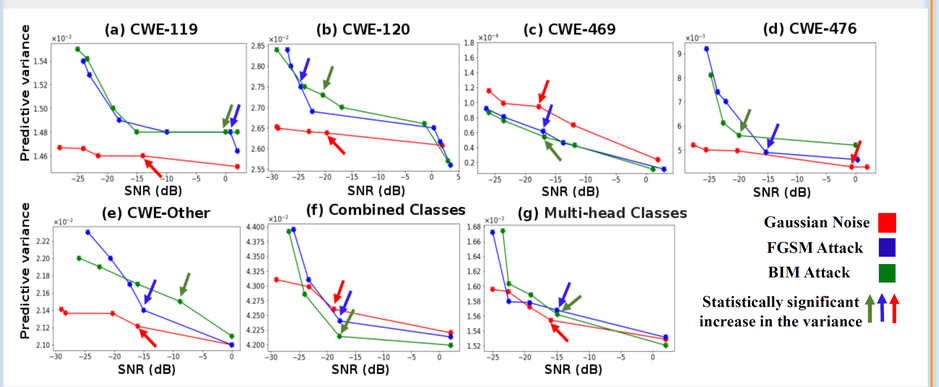

Fig 2: Average predictive variance of different classes plotted against SNR under Gaussian noise, FGSM and BIM adversarial attack.

- ✓ Significant **variance increase** for all classes empowering '**self-awareness**'
- ✓ Higher accuracy with increased noise levels which justifies its '**robustness**'

CONCLUSION

The SVD-GRU model demonstrates **'self-awareness'** and **'robustness'** under high noise levels or stronger adversarial attacks. Such behavior can be used by the model to assess its own performance and alert the user about performance degradation linked to noise or adversarial attacks in high stake applications.

BIBLIOGRAPHY

[1] S. E. Chandy, A. Rasekh, Z. A. Barker, and M. E. Shafiee, "Cyberattack detection using deep generative models with variational inference," Journal of Water Resources Planning and Management, vol. 145, no. 2,p. 04018093, 2019.
[2] H. Dam, T. Tran, T. Pham, S. Ng, J. Grundy, and A. Ghose, "Automatic feature learning for predicting vulnerable software components," IEEE Transactions on Software Engineering, vol. 47, no. 1, pp. 67–85, Jan. 2021.
[3] R. Russell, L. Kim, L. Hamilton, T. Lazovich, J. Harer, O. Ozdemir, P. Ellingwood, and M. McConley, "Automated vulnerability detection in source code using deep representation learning," in Proceedings of the 17th IEEE international conference on machine learning and applications (ICMLA), 2018, pp. 757–762.

Acknowledgments
This work was supported by National Science Foundation Award CRII - 2153413. The authors also gratefully acknowledge the Presidential Research Fellowship (PRF) by UTRGV, the NJ Health Foundation award PC 78-21 and the Texas Advanced Computing Center (TACC - SEE22003).