Improving Website Intrusion Detection Using Similarity Search Vector and Deep Learning Model

1st Sudin Saepudin
Department of Information System
Nusa Putra University
Sukabumi, Indonesia
sudin.saepudin@nusaputra.ac.id

2nd Yansen Makleat
Department of Information System
Nusa Putra University
Sukabumi, Indonesia
yansen.makleat_si20@nusaputra.ac.id

3rd Fauzia Ramadhan
Department of Information System
Nusa Putra University
Sukabumi, Indonesia
fauzia.ramadhan_si20@nusaputra.ac.id

4th Abdul Cholis
Department of Information System
Nusa Putra University
Sukabumi, Indonesia
abdul.cholis_si20@nusaputra.ac.id

***Abstract*—Cyberspace threats are one of the significant issues that information technology based organizations should deal with them. Generally, the security attacks often attempt aimed to gain unauthorized access to the critical data in the information systems and then modify, expose, or use them, the signature-based IDS schemes cannot detect new attacks in which their pattern and signature are unknown. On the other hand, anomaly-based IDS approaches attempt to learn the normal behaviors and recognize everything else as anomaly or intrusion. Nonetheless, they suffer from the false positive problem that restricts their application. This work shows how to use similarity search as a service to improve detection rare events. The datasets were used consist of benign (normal) network traffic and malicious traffic generated from several different network attacks. The Author focused on web attacks only. The web attack category consists of three common attacks, Cross-site scripting (Brute Force-XSS), SQL-Injection (SQL-Injection), and Brute force administrative and user passwords (Brute Force-Web). The result is accuracy for detecting website attacks increased from 29% to 58%. From the overall value, the accuracy of the data that has been used as a similarity search vector has increased from 87.1% to 92.3%.**

Keywords—Intrusion Detection System, Deep Learning, Similarity Search, Web Attack

I. INTRODUCTION

Cyber threats are one of the significant problems that information technology-based organizations must face. Generally, security attacks often aim to gain unauthorized access to critical data in information systems and then modify, expose, or use them [1]. Additionally, some security attacks denoted as Distributed Denial of Service (DDoS) attacks, may attempt to disrupt the normal functioning of a computer system and make it inaccessible to other users and systems. Therefore, in light of the ever-increasing ferocity and diversity of cyber security attacks, providing efficient and effective techniques to exploit them seems crucial.

IDS is one of the main components of security infrastructure that can ward off cyber threats that come from various types of attackers. A wide variety of IDS schemes have been provided in the security literature. In this context, regarding the environments that IDS schemes must protect, they can be classified as host IDS and network IDS approaches [2], where the former intends to secure the computer system by monitoring all events and incoming/outgoing traffic and the latter will monitor and secure the entire computer network. Also, network-based IDS schemes are categorized into flow-based solutions [3], deep packet inspection schemes (Ren et al., 2020) where flow-based approaches only inspect packet header data, but deep inspection schemes. In addition, based on its detection capabilities, IDS approaches are classified as signature detection and anomaly detection approaches [4].

Typically, signature-based IDS benefits from a predefined database of security attack signatures and tries to match events and traffic to specific attack patterns [5]. However, signature-based IDS schemes cannot detect new attacks whose patterns and signatures are unknown. On the other hand, anomaly-based IDS approaches try to learn normal behavior and recognize everything else as an anomaly or intrusion [5]. Nonetheless, they suffer from false positive problems that limit their applications.

This work shows how to use similarity search as a service to improve rare event detection. Such applications are common in the cybersecurity and fraud detection domains where only a small fraction of

events is malicious. The main contributions of this research can be summarized as follows:

- How accurate is deep learning modeling for identifying intrusions on websites.
- How the use of Similarity Search Vectors in datasets improves the accuracy of Deep Learning modeling for detecting website intrusions.

II. LITERATURE REVIEWS

Basnet, et al.[6] applied and compared various state-of-the-art deep learning frameworks (e.g., Keras, TensorFlow, Theano, fast.ai, and PyTorch) in detecting network intrusion traffic and also in classifying common network attack types using CSE-CIC - Latest IDS2018 data set. Experimental results show that fast.ai, a highly opinionated PyTorch wrapper, provides the highest accuracy of around 99% with low false positive and negative rates in detecting and classifying various types of intrusions. The results provide evidence of the usefulness of various deep learning frameworks that detect network intrusion traffic.

Komarek, et.al.[7] performed a similarity search method (called Random Split) that helps threat analysts by identifying unknown variants of known malware in network traffic. This method assumes that for each malware family being hunted, only a few network communication samples are available to the analyst (multi-positive) and the others are hidden in the abundant network data (unlabeled). The authors demonstrate the method on large-scale real-world data, which outperforms unsupervised approaches (Isolation Forest and Lightweight Online Anomaly Detector), supervised approaches (Random Forest) and traditional similarity search algorithms (kNN). This evaluation involved eight high-risk malware families with varying known/unknown ratios.

Gomathy, et.al.[8] proposed a vector quantization approach based on a supervised deep learning approach to detect Memcached attacks carried out using malicious firmware on various types of Cloud attached devices. This vector quantization approach detects DDoS attacks carried out by malicious firmware on various types of cloud devices and it also classifies applications that are vulnerable to attacks based on cloud services-The Hackbeased. The results calculated during testing showed 98.2% legal positives and 0.034% false negatives.

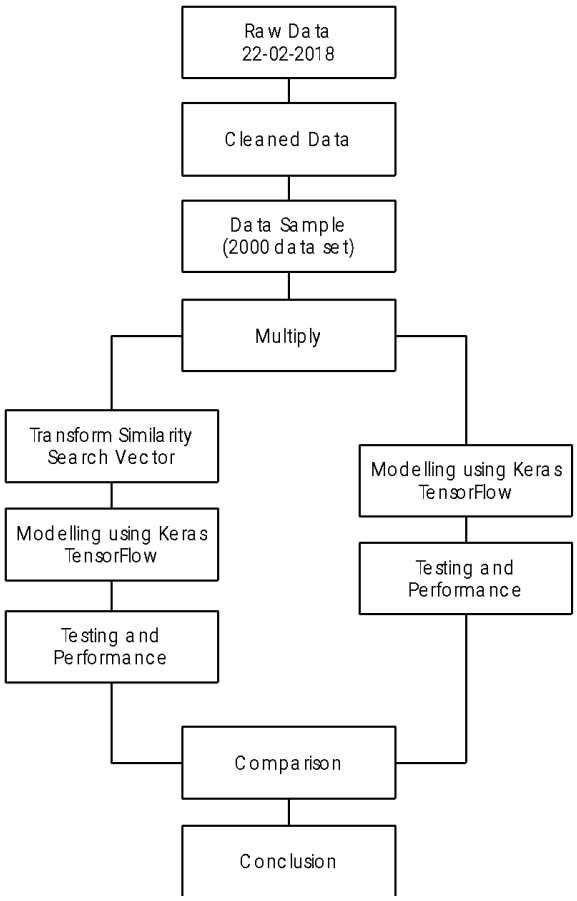
Jiang Li[4] combines the known deep learning related literature in the field of intrusion detection, and the existing intrusion detection system design based on deep learning general low detection efficiency and high false positive rate, as well as the training dataset used by the problem of data that is not balanced, etc., through in-depth study of feature extraction and data processing capabilities. To study Internet of Things intrusion detection methods based on deep learning. The work and achievements are as follows: First of all, it will be used for feature selection of embedded models (EM) and used in convolutional neural network (CNN) intrusion detection, combined with lightweight intrusion detection models (XCNN), because it is an embedded model (EM), so as to reduce the impact on equipment, The effectiveness of the method is proven by building an Internet of Things environmental laboratory for actual testing. Experimental results show that the algorithm can greatly reduce training time, improve training efficiency, and have the same or even better performance results.

Takeda and Nagasawa[9] created a deep learning method for intrusion detection systems that can detect U2R and R2L attacks with higher accuracy than existing methods. The proposed method only uses simple deep learning techniques such as convolutional neural networks, over-sampling, under-sampling and data augmentation. Moreover, the proposed method does not require any domain knowledge, as the proposed pipeline does not involve feature engineering. In this paper, we also present experimental results evaluating the performance of the proposed method using the KDD Cup 99 Dataset. The experimental results show that the proposed method can detect U2R or R2L attacks with higher accuracy than previous studies.

Otoum, et al.[10] present a comprehensive analysis of the use of machine and deep learning (DL) solutions for IDS systems in wireless sensor networks (WSN). To achieve this, we introduce restricted Boltzmann machine-based IDS (RBC-IDS), a potential DL-based IDS methodology for monitoring critical infrastructure by WSNs. We study the performance of RBC-IDS, and compare it with a previously proposed adaptive machine learning-based IDS: adaptive supervised and clustered hybrid IDS (ASCH-IDS). Numerical results show that RBC-IDS and ASCH-IDS achieve similar detection rates and accuracy, although the detection time of RBC-IDS is approximately twice that of ASCH-IDS.

III. RESEARCH METHODOLOGY

Experiment Overview



Network intrusion detection systems monitor the flow of incoming and outgoing network traffic, sounding an alarm whenever a threat is detected. Here we use deep learning models and similarity search in detecting and classifying network intrusion traffic.

We will start by indexing a set of labeled traffic events in the form of vector embeddings. Each event can be benign or malicious. Rich vector embeddings, mathematical representations of network traffic events. This makes it possible to determine how similar network events are to each other using the similarity search algorithm built into Pinecone.

A. Data Preparation

The authors downloaded and used the latest network intrusion dataset generated and published in 2018. For more information about the dataset, including the experiments and testbed used to generate the dataset. The dataset consists of both benign (normal) network traffic and malicious traffic resulting from several different network attacks, primarily those classified as web attacks.

B. Data Pre-processing

After downloading the dataset, we analyze the data to study its characteristics and clean it if necessary. The data set consists of the original traffic in the pcap file, the received logs and labels, and the selected CSV file displays. We use a CSV labeled file with a total of 80 traffic features extracted using CICFlowMeter. The dataset includes a cross section of normal (benign) and attacks consisting of various common attack types—stated in the previous section—distributed among seven CSV files. Due to the large number of samples available and also to keep the experiment simple, we dropped samples with Infinite Values, NaN, or missing values.

C. Similarity Search Vector

Most tools, deep learning algorithms or [11] statistical ML methods operate with initial item representations in vector form in a high-dimensional space, called embeddings. This makes it possible to represent the initial item (such as text, image, etc.) in a much more efficient and flexible way while also saving its features and sometimes even its context. Basically, looking for similar items means looking for their vector representations that are close to each other in their representation space (finding the Euclidean or other distance between them).

In this experiment, the author will upload the data set to Pinecone.io. Pinecone is a fully managed vector database that makes it easy to add vector search to production applications. It combines advanced vector search libraries, advanced features such as filtering, and distributed infrastructure to provide high performance and reliability at any scale.

D. Keras TensorFlow Deep Learning

Keras Interface is a portable, high-level neural network API, written in Python, usable for TensorFlow, CNTK, and Theano as a back-end, and was originally developed as part of the Open Neuro-Electronic Intelligent Robot Operating System (ONEIROS). This interface has many advantages in research and development; mainly because of its portability. Because Keras is written to support three major deep learning frameworks and potentially more in the future, minimal changes are required to replace the framework used.

TensorFlow is an open-source machine learning platform developed by Google with wide industrial use. It has a comprehensive and flexible ecosystem of tools, libraries, and community resources that enable researchers to drive advanced machine learning (ML) and developers to easily build and deploy ML-powered applications. This framework is available in Python, Java, JavaScript, and C++, while also supporting Internet of Things (IoT) devices.

E. Measurement

There are several metrics available to evaluate and compare the performance of machine learning classifiers. The authors use the following performance metrics extensively in literature to evaluate deep learning classifiers.

- *Accuracy*

  Accuracy is the percentage of samples correctly classified over the total number of samples evaluated. Accuracy by itself may not be a good measure of performance when the number of samples evaluated is disproportionate. Therefore, the authors also created and reported confusion metrics that address the shortcomings of accuracy measures.

- *Confusion matrix*

  The confusion matrix provides a visual way to display the detailed performance of a classifier showing the total number of correctly and incorrectly classified samples in each category. The confusion matrix also helps us calculate the false positive rate—also called recall or sensitivity which is the % of negative class samples that are classified as positive class and false negative rate which is the % of positive class samples that are incorrectly classified as negative—other important metrics used in classification problems that can further be used to calculate commonly used measures such as Precision and F1 measure. Confusion matrices are very useful in evaluating not only binary class classifiers, but also multi-class classifiers as in our case

IV. RESULTS AND DISCUSSION

The dataset we use consists of both benign (normal) network traffic and malicious traffic resulting from several different network attacks. We will focus on web attacks only. The web attack category consists of three general attacks, Cross-site scripting (Brute Force-XSS), SQL-Injection (SQL-Injection), and administrative and user password brute force (Brute Force-Web). The original data was recorded over two days on 02-22-2018.

The authors cleaned the data by removing missing rows, then separating all labels that were threats, resulting in 5,610 rows being removed from the data.

TABLE I. Data Explanation

-	Name	Label	Count	Drop
	02-22-2018.csv	Benign	1,042,603	
		BruteForce-Web	249	
		BruteForce-XSS	79	
		SQL-Injection	34	
	02-22-2018.csv	Benign	1,042,603	5610
		BruteForce-Web	249	0
		BruteForce-XSS	79	0
		SQL-Injection	34	0

TABLE II. CLEANED DATA

Benign	1042603
Brute Force -Web	249
Brute Force -XSS	79
SQL Injection	34

Name: Label, dtype: int64

From the data that had been cleaned, 2,000 data were then selected to be used as sample data in this research. The 2,000 records include all samples containing attacks.

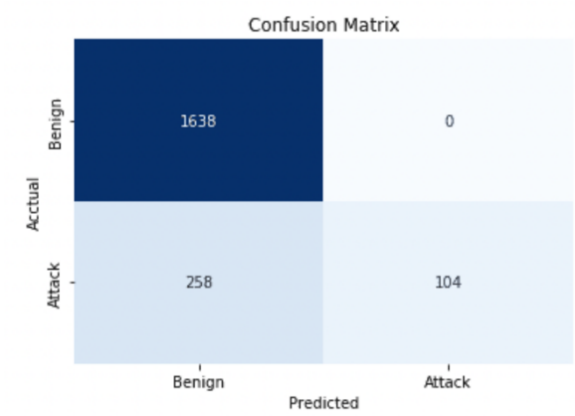
TABLE III. EXAMPLE DATA

Benign	1638
Brute Force -Web	249
Brute Force -XSS	79
SQL Injection	34

Name: Label, dtype: int64

The author changed the number of classes from four (Benign, Brute Force-Web, Brute Force-XSS, SQL-Injection) to two (Benign and Attack). Next, experiments were carried out using deep learning on selected sample data, and the following confusion matrix was obtained:

CONFUSION MATRIX ORIGINAL DATA



From the Confusion Matrix that is formed, it can be seen that the deep learning model used is able to detect all data that is included in the Benign class. Meanwhile, for the Attack class, this model was only able to detect 104 data, while 258 data were classified as Benign class.

OVERALL PERFORMANCE

```
Accuracy: 0.871
Precision: 1.000
Recall: 0.287
```

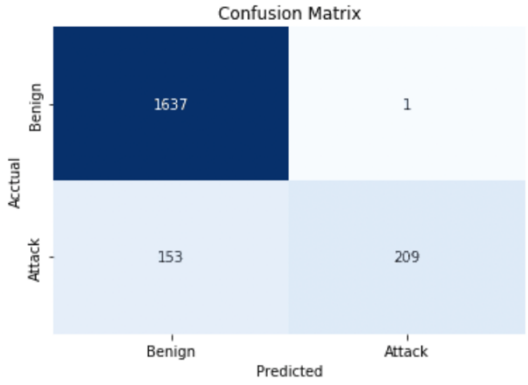
CLASS PERFORMANCE

	type	accuracy
0	Benign	1.00
1	Attack	0.29

The accuracy obtained using this method was 87.1%, with a recall of 0.287. Next, accuracy calculations were carried out based on class, and the results obtained for data classified as benign reached 100%, while the accuracy value for attack data reached 29%.

The second experiment was carried out by transforming sample data into search vectors on the Pinecone service. The resulting data is then entered into deep learning modeling and the following confusion matrix is obtained:

CONFUSION MATRIX SIMILARITY SEARCH FOR VECTOR DATA



The results of the deep learning model operated on vector data similarity searches can detect all data that is included in the Benign class. Meanwhile, for the attack class, this model was able to detect 209 website attacks, and 153 other pieces of data which were classified as benign.

OVERALL PERFORMANCE

```
Accuracy: 0.923
Precision: 0.995
Recall: 0.577
```

CLASS PERFORMANCE

	type	accuracy
0	Benign	1.00
1	Attack	0.58

The accuracy value produced in the experiment using vector data search reached 92.3% with a recall value of 0.577. Meanwhile, for the performance value per class, the benign class reaches 100% while the attack class reaches 58%.

From the comparison of the two experiments that have been carried out, the accuracy value for detecting website attacks increased from 29% to 58%. Meanwhile, benign data remains the same with 100% accuracy. Of these overall values, the accuracy value of the data that has been used as a similarity search vector increased from 87.1% to 92.3%.

V. CONCLUSION

From the comparison of the two experiments that have been carried out, the accuracy value for detecting website attacks increases. Experiments show that the similarity search approach outperforms direct classification approaches that utilize classifier embedding models. Similarity search-based detection achieves 50% higher accuracy compared to direct detector.

ACKNOWLEDGMENT

The Author thanks to the Lecturer and Rectorate of Nusa Putra University for supporting this research. for providing a conducive academic environment and resources that have greatly contributed to the realization of this research. The university's commitment to excellence in education and research has undoubtedly played a pivotal role in my academic journey.

REFERENCE

[1] P. A. Sonewar and S. D. Thosar, "Detection of SQL injection and XSS attacks in three tier web applications," *2016 International Conference on Computing Communication Control and automation (ICCUBEA),* pp. 1-4, 2016.

[2] A. Almalawi, Z. Tari, A. Fahad and X. Yi, "A Global Anomaly Threshold to Unsupervised Detection," *SCADA Security: Machine Learning Concepts for Intrusion Detection and Prevention,* pp. 119-149, 2021.

[3] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras and B. Stiller, "An Overview of IP Flow-Based Intrusion Detection," *IEEE Communications Surveys & Tutorials,* vol. 12, no. 3, pp. 343-356, 2010.

[4] L. Jiang, "Research on Intrusion Detect System of Internet of Things based on Deep Learning," *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE),* pp. 55-58, 2022.

[5] S. K. Patnaik, C. N. Babu and M. Bhave, "Intelligent and adaptive web data extraction system using convolutional and long short-term memory deep learning networks," *Big Data Mining and Analytics,* vol. 4, no. 4, pp. 279-297, 2021.

[6] R. Basnet, R. Shash, C. Johnson, L. Walgren and T. Doleck, "Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks," *Journal of Internet Services and Information Security,* vol. 9, no. 4, pp. 1-17, 2019.

[7] T. Komarek, J. Brabec, C. Skarda and P. Somol, "Threat Hunting as a Similarity Search Problem on Multi-positive and Unlabeled Data," *2021 IEEE International Conference on Big Data (Big Data,* pp. 2098-2103, 2021.

[8] S. Gomathi, N. Parmar, J. Devi and N. Patel, "Detecting Malware Attack on Cloud using Deep Learning Vector Quantization," *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN),* pp. 356-361, 2020.

[9] A. Takeda and D. Nagasawa, "A Simple Deep Learning Approach for Intrusion Detection System," *021 Thirteenth International Conference on Mobile Computing and Ubiquitous Network (ICMU),* pp. 1-2, 2021.

[10] S. Otoum, B. Kantarci and H. T. Mouftah, "On the Feasibility of Deep Learning in Sensor Network Intrusion Detection," *IEEE Networking Letters,* vol. 1, no. 2, pp. 68-71, 2019.

[11] S. K. Patnaik, C. N. Bhabu and M. Bhave, "ntelligent and adaptive web data extraction system using convolutional and long short-term memory deep learning networks," *Big Data Mining and Analytics,* vol. 4, no. 4, pp. 279-297, 2021.