CHAPTER I
INTRODUCTION

1.1 Background

Internet is a technology that allows computers to communicate with other computers around the world [3]. Internet is built with numerous networks that contain various devices such as routers and switches [4]. Those devices utilize the proprietary Operating System (OS) or command that hinders the integration for a different vendor. For example, vendor A required some policies and requirements to connect to vendor B that required to downgrade of the OS, which may affect the connection to vendor C. Furthermore, the difference between the operating system and command will make the network administration more difficult because the maintenance will take more time to figure out.

Based on this problem, researchers figure out the solution by creating a network that can be programmed namely a Software Defined Networking (SDN). The SDN has three characteristics [5], which are: (1) the data plane of the network (packet forwarder) is separated from the control plane (routing decision) as shown in Fig 2.1, (2) the control plane can handle multiple data plane via one console, (3) the network administrator can handle multiple networks simultaneously. Even though the SDN has advantages for maintaining the network, it still has similar vulnerabilities as the traditional network. Those vulnerabilities affect the Confidential, Integrity, and Availability (CIA) on the SDN network as described in works [6] and [7].

To overcome those issues, the Intrusion Detection System (IDS) is required to detect any anomaly in the network traffic such as a cybersecurity attack. There are two kinds of IDS, the first is Host IDS (HIDS) and the second is Network IDS (NIDS). The NIDS is the IDS deployed in user local networks and utilized for monitoring the traffic flows in this network. The NIDS has two detection approaches, namely the signature-based and anomaly-based [6]. The signature-based NIDS generates an alert when the traffic flow has the same pattern as the rule set. Meanwhile, the signature-based detection approaches are effective to recognize common attacks on the network, but they cannot adapt to new network properties, such as a new signature. An example of NIDS with signature-based approaches is Snort. The rule on the signature-based is regularly updated after any new attack is discovered. Meanwhile, anomaly-based detection works when any anomaly or deviation appears in

the network.

To enhance the detection approaches, several researchers employ the Artificial Intelligence (AI) approaches to the NIDS as discussed in works [8], [9]. Kanimozhi et al. [8] discussed using Artificial Neural Networks (ANN) on the CIC dataset with a cloud computing platform. The cloud computing platform gives two advantages, like scalability computing resources and the ability of high-scale data processing to make real-time traffic detection possible. Meanwhile, Lansky et al. [9] compared the algorithm on AI, such as Auto Encoder (AE), Restricted Boltzmann Machine (BM), Deep Belief Network (DBN), Recurrent Neural Network (RNN), Deep Neural Network (DNN), and Convolutional Neural Network (CNN). Most of these algorithms are applied in NSL-KDD [10] dataset which consists of 41 features. The accuracy output of the method varies from 0.79 to 1. They explain two major issues regarding the implementation of the AI method in IDS, which are: (1) the requirement is high to train the IDS dataset with AI, and (2) some datasets are difficult to classify due to the imbalance of data between normal and anomaly.

To overcome these problems, we create lightweight data pre-processing to balance the InSDN dataset [2]. Besides that, we use feature correlation to reduce the complexity of the dataset. This research also compares two data-to-image conversion methods. First is the two-channel color method [11] and the second is the grayscale method. These images will be trained with Convolutional Neural Network (CNN) to classify the vulnerability in the InSDN dataset.

1.2 Problem Identification and Objective

To enhance the anomaly-based IDS, we propose a CNN method to classify the image that contains various features from the InSDN dataset This algorithm consists of data pre-processing, data-to-image conversion, and CNN learning process. The downside of the CNN method is the high computational resource [12]. Besides that, the InSDN dataset still has an extremely imbalanced dataset that affects the classification metrics. Based on these problems, there are several objectives that need to achieve in this research :

1. Make a lightweight data pre-processing to balance the dataset and reduce the complexity of the dataset,
2. Evaluate image conversion methods for CNN using accuracy, F1-score, precision, and recall by configuring the size of the dataset, the feature size, and the image conversion method (Two-channel, and grayscale),
3. Analysis of the effect of conversion and learning process on the host by com-

paring the total time.

1.3 Related Research

There are several research for the CNN algorithm with a data-to-image conversion method. Sun et al. [13] converted the dataset row into a matrix image only. This method called SuperTML consists of two methods which are using variable font size based on the feature importance and using equal font size. Compared with XGboost [14] and DNN, the SuperTML has an advantage with higher accuracy. The downside of this method is still only applicable to the iris dataset.

Kim et al. [15] implemented a CNN algorithm with KDD CUP 1999 [16] and CSE-CIC-IDS2018 [17] datasets. In this research, the authors employed one-hot encoding to classify the attack in the dataset. The image model is generated by converting the tabular datasets into a matrix. The size of the matrix depends on the feature numbers of the datasets. The matrix is filled with data from each row from the table and normalized with a maximum value of 255. Furthermore, all the values filled in the matrix were converted to color. The color scheme was divided into two scenarios, the first is RGB and the second is grayscale. Based on this simulation, the author concluded that the best performance of the CNN algorithm while using the RGB and the grayscale was by using 2×2 and 3×3 kernel size.

In 2022, researchers [11] and [18] conducted IDS simulations with image vision in a wireless network. The dataset used in the simulation is AWID2. In [11], the authors applied color-based image vision, where the dataset is converted into two colors before the CNN learning method. As for work [18], the authors applied a matrix-based image where the dataset was converted to a matrix with a zig-zag pattern before the learning method. The simulation results in [11] and [18] show that the F1-score on color based has greater value compared to the matrix one or compared to the signature-based learning method. Even though the color-based approach showed a better F1 score than the matrix image, the period of data processing took about an hour.

Based on this research the data-to-image conversion method with the CNN learning method can boost the classification metrics compared to regular machine learning. Afterward, our research will add several methods in data pre-processing for the InSDN dataset before applying the data-to-image conversion process. This method includes the dataset class balancing and feature selection to improve the classification metrics and reduce the CNN learning time.

1.4 Scope of Work

To limit the discussion on this topic, we describe several scopes of work for this research:

1. The data balancing method used in this research is Synthetic Minority Oversampling Technique (SMOTE),
2. Pearson correlation will be used for the data correlation process,
3. The data-to-image conversion method used in this research is two-toned color and grayscale.
4. The deep learning method used in this research is CNN with TensorFlow backend,
5. Compared the CNN method with Artificial Neural Network (ANN) method, and Random Forest (RF),
6. Evaluate the effect of the number of data and the number of features.

1.5 Research Methodology

These are some methods that need to work, to complete this research:

1. Study Literature
   This process is a learning stage about the theories of the SDN, CNN, and IDS from the newest source such as papers, journals, and books,
2. Design System Model
   Design a system model based on the study literature. The model consists of dataset pre-processing, dataset conversion to image, CNN training process, and the output of the learning process,
3. Simulation
   This process starts with getting the dataset, processing the dataset, converting the dataset into images, and training the data with CNN
4. Result and Analysis
   Analyze the result achieved in simulation processes. This analysis contains classification metrics, learning time, and confusion matrices.
5. Conclusion
   The result and analysis are used to conclude the research problems and purposes that have been stated before.

1.6 Research Timeline

The research timeline can be seen in Table 1.1 and the output will be published from November to December.

Table 1.1 Research timeline.

Timeline / Activity	July	August	September	October	November	December
Create system model in Jupyter						
Build data preprocessing						
Build tabular data to image conversion						
Continue the paper/journal Section III						
Train the CNN						
Continue the paper/journal Section IV						
Publish Paper						
Paper Review and Acceptance						
Conclusion						

1.7 Structure of The Thesis

The rest of this thesis is organized as follows:

- **CHAPTER 1: INTRODUCTION**
  This chapter discusses the background of this research, from the problem in the field, related research, the scope of work, and the research methodology that use in this research

- **CHAPTER 2: BASIC CONCEPTS AND PROPOSED CNN MODEL**
  This chapter provides basic information for this thesis, including an explanation of SDN, IDS, the InSDN Dataset, the data-to-image conversion method, and the CNN method that will be applied to this research

- **CHAPTER 3: SYSTEM MODEL AND RESEARCH DESIGN**
  This chapter describes the system model including parameters and variables used in the thesis, research flow, and how the simulation works in the algorithm.

- **CHAPTER 4: PERFORMANCE EVALUATIONS**
  This chapter discusses the result of this thesis, starting from the data-to-image conversion methods and the output of the simulation that consists of classification metrics, confusion matrix, and the learning process time
- **CHAPTER 5: CONCLUSIONS AND FUTURE WORKS**
  This chapter provides the conclusion of this thesis and the recommendation for future works.