JOIV : Int. J. Inform. Visualization, 6(2) - June 2022 306-316

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

An Intrusion Detection System Using SDAE to Enhance Dimensional Reduction in Machine Learning

Hanafi [a,*], Alva Hendi Muhammad [a], Ike Verawati [a], Richki Hardi [b]

[a] *Department of Computer Science, University of Amikom Yogyakarta, Depok, Sleman, 55283, Indonesia*
[b] *Departement of Informatic, Universitas Mulia, Balikpapan, East Kalimantan, 76114 , Indonesia*
*Corresponding author: *hanafi@amikom.ac.id.ac.id*

***Abstract*—In the last decade, the number of attacks on the internet has grown significantly, and the types of attacks vary widely. This causes huge financial losses in various institutions such as the private and government sectors. One of the efforts to deal with this problem is by early detection of attacks, often called IDS (instruction detection system). The intrusion detection system was deactivated. An Intrusion Detection System (IDS) is a hardware or software mechanism that monitors the Internet for malicious attacks. It can scan the internetwork for potentially dangerous behavior or security threats. IDS is responsible for maintaining network activity under the Network-Based Intrusion Detection System (NIDS) or Host-Based Intrusion Detection System (HIDS). IDS works by comparing known normal network activity signatures with attack activity signatures. In this research, a dimensional reduction and feature selection mechanism called Stack Denoising Auto Encoder (SDAE) succeeded in increasing the effectiveness of Naive Bayes, KNN, Decision Tree, and SVM. The researchers evaluated the performance using evaluation metrics with a confusion matrix, accuracy, recall, and F1-score. Compared with the results of previous works in the IDS field, our model increased the effectiveness to more than 2% in NSL-KDD Dataset, including in binary class and multi-class evaluation methods. Moreover, using SDAE also improved traditional machine learning with modern deep learning such as Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). In the future, it is possible to integrate SDAE with a deep learning model to enhance the effectiveness of IDS detection.**

***Keywords*—IDS detection; SDAE; naive bayes; decision tree; SVM; auto encoder.**

Manuscript received 5 Jan. 2022; revised 11 Mar. 2022; accepted 20 Apr. 2022. Date of publication 30 Jun. 2022.

I. INTRODUCTION

The number of internet users has increased significantly over the last decade. Additionally, advancements in technology, particularly in the internet, communication, and networking, have resulted in a massive amount of data being generated from a variety of sources, including industry, e-commerce portals, messengers, social media, and healthcare. This massive amount of data is referred to as big data and has four characteristics: high veracity, high velocity, wide variety, and high value. Since the advent of big data, the number of attacks has also increased. In 2019, the internet had been connected to more than 26 billion devices. Additionally, it contributes to the growth of malicious activity on the internet. Intrusion Detection System (IDS) has evolved into a critical tool for enhancing network and computer system security [1], [2].

Numerous experts, researchers, and academicians use conventional machine learning mechanisms to improve IDS, including Neural Networks (NN), Support Vector Machines (SVM), K Nearest Neighbors (KNN), Decision Tree 3 (DS3), Multi-Layer Perceptron (MLP), and Auto Encoder (AE). The involvement of conventional shallow learning frameworks (one feedforward network) is ineffective in resolving the autodetection problem for big data. They consistently fail to detect activity attacks, accurately capture attack information, and resolve noise in massive datasets [3], [4]. In response to the issue above, deep learning models such as a deep Auto Encoder (AE), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) have become increasingly popular in recent years. The illustration of IDS detection is shown in Fig. 1 [5].

Additionally, the total number of attributes extracted from the internet data that IDS must observe is always enormous, even in small-scale capacity networks. Indeed, the majority of raw data is superfluous and noisy. As a result, the classifier's performance is degraded by the presence of unsuitable features. As a result, it is critical to employ multidimensional

reduction frameworks such as the Principal Component Analysis (PCA), Mutual Information (MI), Chi-square, and UMAP [6]. Unlike the previous works, our experiment adopted SDAE to enhance dimensional reduction. The detailed experiment scenario is shown in Fig. 2.

In this study, the researchers developed a novel dimensional reduction model based on SDAE, focusing on four aspects, including 1) the hybridization between SDAE and KNN, 2) the hybridization between SDAE and Naive Bayes, 3) the hybridization between SDAE and SVM, and 4) the hybridization between SDAE and decision tree. We have applied the proposed model mentioned above to the NSL-KDD dataset.

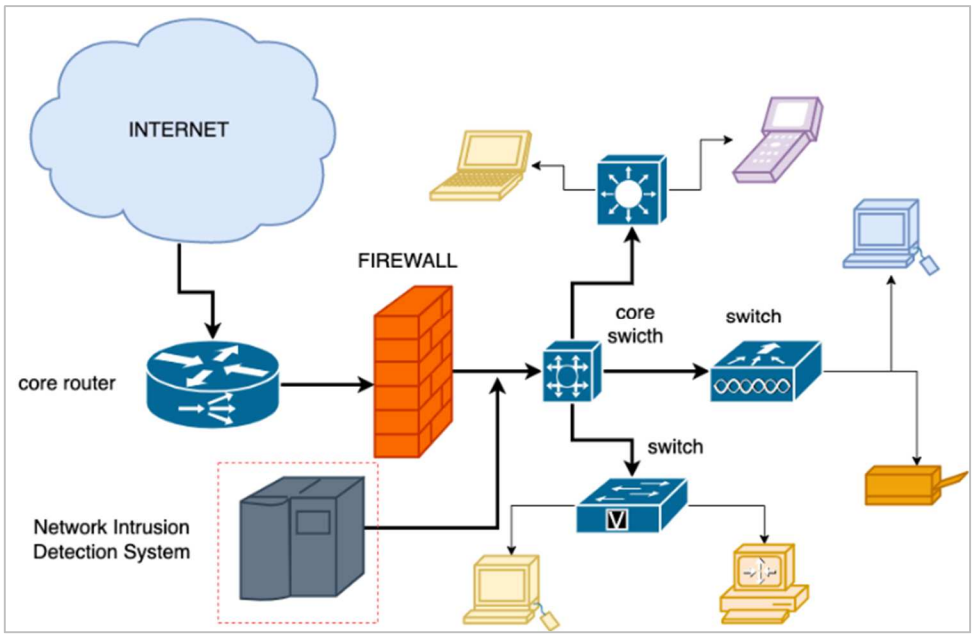

Fig. 1 IDS detection illustration

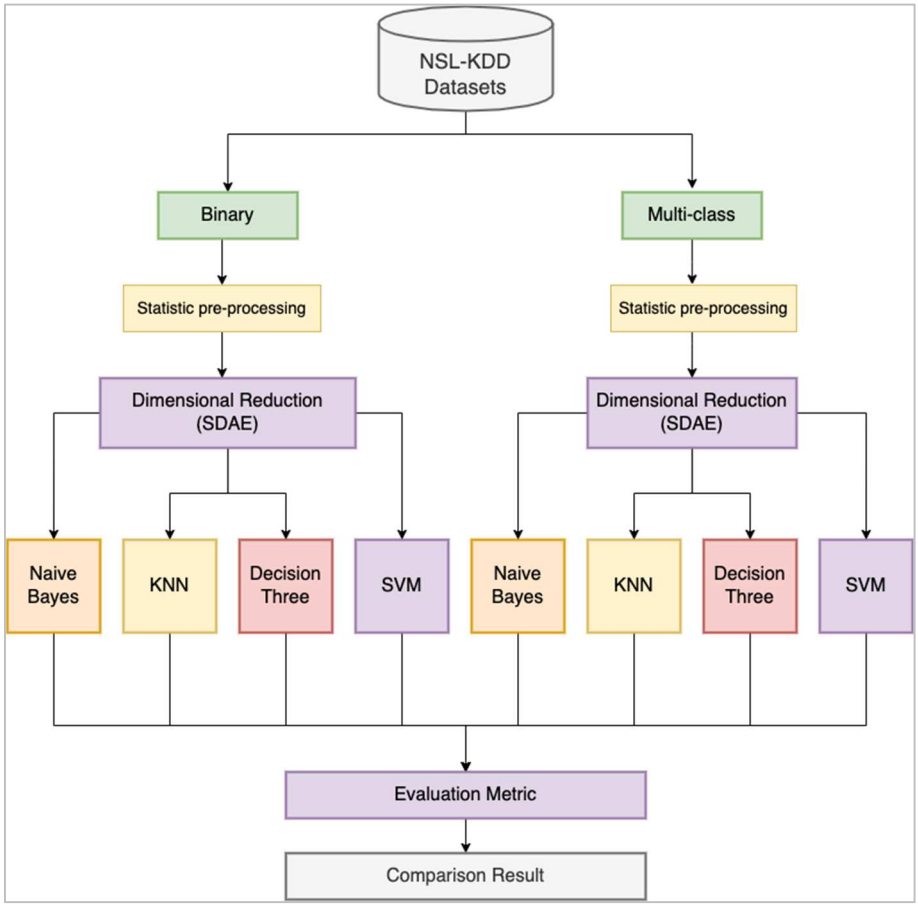

Fig. 2 Experiment scenario of IDS detection

Many previous works state that the intrusion detection model has three main methods: deep learning, conventional machine learning, and pattern similarity. Deep learning has become the most popular method in the last few years. In the beginning, pattern similarity models were mostly used to detect intrusions. Most of them use patterns similar to their main core learning algorithm, and they use attribute similarity to do this [7], [8]. Most of the frameworks have already been used for implementation in the past. Knuth Morris Pratt (KMP), Boyer Moore (BM), Boyer Moore Harspool (BMH), Boyer Moore Harspool Sunday (BMHS), Aho-Corasiek (AC), and AC-BM were some of the traditional models that were used to make an Intrusion Detection System. Following the experiments' results, it was found that an algorithm worked well to speed up the performance of pattern similarity calculations and cut down on the amount of time it took to do them. However, the traditional pattern similarity model has a big problem. They cannot figure out how intrusion detection works. The discovery of a low-cost algorithm that can cut down on the amount of time it takes, and the value of false positives has become the main point of this study. When machines become more intelligent, there is still a new study that is worth reading.

Denning [9] was the first to propose IDS machine intelligence, and his study used a multi-algorithm model to detect intrusion detection activity. According to the expert hypothesis, the model created a pattern of several features by hand. First, a modern machine learning model based on SVM was created [10]. The experiment configured KDD99 datasets, resulting in 3 features with an accuracy of 91%, 36 features with an accuracy of 99%, and 41 features with an accuracy of 99%.

A study employing traditional machine learning and KNN improved an early model. This model included a K-mean clustering and a KNN classifier [12]. This model evolved into the state-of-the-art IDS intelligence machine for malicious detection known as CANN. Another study proposes the use of a traditional classifier with Random Forest to improve CANN [11]. The hybrid model, which used Random Forest as a core classifier machine, achieved an accuracy of 94.7%. A Random Forest (RF) enhancement using an Artificial Neural Network (ANN) was proposed [12]. When applied to NSL-KDD, the ANN model produced more than 81% of accuracy and 79% classification for malicious detection and network attack classification. A Decision Tree (DT) intrusion detection model based on NSL-KDD was proposed [13]. According to the experiment results, DT successfully achieved effectiveness in the IDS detection classification task. According to the explanation given above, the enhancement of traditional machine learning achieves astounding effectiveness in IDS detection. However, most of them required large-scale pre-processing and complex attribute extraction. It is impossible to handle significant intrusion data when using a machine learning classification method.

Deep learning, a new type of neural network with a very complex network structure, was introduced in the early decade. Deep learning had achieved tremendous performance in the image processing classification task. Furthermore, deep learning has become the industry standard for dealing with a variety of computer science-related problems such as image processing, voice recognition, text mining, recommender system [14], [15], [16], [17], [18], [19], matrix factorization enhancement for recommender system [20]. Recommender system based on location for transportation service [21], product document representation to enhance collaborative filtering based on matrix factorization [22], [23], [24], CNN for document context for recommender system [25].

A deep learning model based on Auto Encoder was proposed [26], using NSL-KDD to investigate the self-taught learning model (STL). The model is made up of two fundamental process classifications. The first step in the compact attribute representation process is to train a dataset with unlabeled data, and the second process is to train the learning representation features with labeled data and implement the classification of IDS tasks. The experiment used STL in two, five, and twenty-three classes. According to the results, STL achieved an accuracy of 88.39%, while the 5-class classification achieved an accuracy of 79.10%.

A deep learning model was based on the combination of Deep Belief Networks (DBNs) and probabilistic neural networks [27]. DBN is responsible for converting low-dimensional to non-linear representations while retaining the important characteristics of raw data. They optimize hidden layer learning using particle swarm optimization. Additionally, the Probabilistic Neural Network (PNN) uses final classification techniques for IDS detection. As demonstrated in their experiment, DBN-PNN achieved an accuracy of 93.25%. Additionally, DBN-PNN outperformed previous works that combined Principal Component Analysis (PCA) and Probabilistic Neural Networks (PNN).

A study proposed another deep learning model for the IDS task based on a Deep Belief Network (DBN) [28][29]. This model incorporates two critical processes: 1) they learned layer by layer using a restricted Boltzmann Machine (RBM), and 2) they derive the hidden layer vector from the visible layer vector. The hidden layer representation is the vector manifest for the following layer. The two processes combine backpropagation networks generated by the final RBM method and use the output vector generated by RBM as an input vector. The DBM model achieves a measurement accuracy of 95.25%. This results in a performance advantage of 89.07% over backpropagation and 91.36% over SVM.

DNN is an acronym for Deep Neural Network, considered suitable for use in IDS networks [30]. The DNN algorithm represents an auto encoder with four hidden layers and one hundred hidden units. They use Rectified Linear Units (ReLU) to activate the hidden layer, and ReLU classifies activation functions that are not linear. This activation function is intended to improve the algorithm's performance when performing complex classification tasks. The adaptive moment mechanism was used in this study to reach the stochastic optimizer. As demonstrated in the experiment, DNN achieved a measurement accuracy of 99%.

A novel model for detecting IDS networks using Convolutional Neural Networks (CNN) has been proposed [31]. The CNN model is well-suited to address a variety of image processing-related issues. In this IDS detection case, the author assumed that the image processing problem is similar to the IDS problem in terms of data vector dimension. CNNs are a subclass of feedforward neural networks that employ convolutional processes to condense large amounts of dimensional data into representative vectors. This work,

which employs a CNN model, asserts that the model successfully improved the imbalanced dataset and that the model not only reduced the false alarm rate but was also useful in enhancing the class's accuracy even when the sample size was small. As their experiment report indicates, CNN achieves an accuracy of 79.48% in KDD-NSL. It outperforms several conventional machine learning techniques that have been proposed in previous works.

GAN (Generative Adversarial Network) and AE were used on NSL-KDD, a novel IDS detection model [32]. When they applied a semi-supervised model, they reduced the time and effort required to manually label the labeled data and increased the effectiveness of IDS malicious detection without labeled data. Using GANs and AEs to improve IDS detection on NSL-KDD datasets, even with only 0.1% of the datasets that had labeled data, was a successful experiment report.

The Long Short-Term Memory (LSTM) is a subclass of feedforward neural networks with sequential aspect mechanisms [33], [34]. It is a recurrent neural network enhancement. This year, LSTM is being considered a possible model for an IDS network, such as the so-called DL-IDS [26]. DL-IDS has an accuracy rate of 98.67%, according to an experiment on Hybrid PCA/LSTM [35]. PCA is responsible for reducing raw data attack dimensions, while LSTM is tasked with classifying network attacks. They report that PCA-LSTM achieves 99.45% accuracy in binary class and 99.39% accuracy in multiclass. LSTM performance was improved by reducing the number of dimensions in the PCA model. They also proposed mutual information (MI) and LSTM in their research. It has a 96.24% binary class accuracy and a 95.56% multi-class classification accuracy.

II. Material and Method

This study considers using NSL-KDD datasets to assess the efficacy of SDAE KNN, SVM, and Decision Tree variants. The datasets are widely used in IDS detection research. The detailed explanation and representative datasets are provided below.

A. NSL-KDD datasets explanation

NSL-KDD is an improved version of the KDD99 datasets. The datasets are widely used in the benchmarking mechanism of many IDS network detection systems. Furthermore, NSL-KDD improves some shortcomings in the original KDD99 datasets, such as the lack of repetition and replication in test and train records, which influences the bias of the classifier function against frequent samples. The dataset was created for free use by the Canadian Cybersecurity Institute [36]. The datasets are divided into training and testing configurations, which are denoted as KDD_{Train+} and KDD_{Test+}, respectively, with a total of 125973 training records and 22544 testing records. Begun in the KDD_{Test+} recognized with additional 17 attack categories, in which it is not integrated into KDD_{Train+}, the researchers aim to achieve a classification result fairly, and thus removing 3751 categories was considered necessary. Furthermore, the KDD_{Test+} was 22544 - 3751 = 18793. Table 1 shows the detailed characteristics of the KDD_{Train+} and KDD_{Test+}. NSL-KDD, including the z_f (f=1,2,3,4,5,..41) feature, which includes three symbolic attributes and 38 continuous attributes. The NSL-KDD datasets are divided into four attack class categories, as described below:

- Denial of Service (DoS): A DoS attack is when someone tries to make it impossible for people to get to a network service, server, or other services by flooding the internet with a lot of traffic. In a DoS attack, someone else can slow down or shut down a server or network service.
- Root to Local (R2L): R2L attacks send remote packets that are not real to a server or computer system to get into the server or computer system without permission.
- User to Root (U2R): It is a group of attacks to get into a computer's "root" area. In this example, the hacker finds out the system's flaw and logs in as a normal person.
- Probe: It is an attack category that can get information about networks and security management systems without being under the control of anyone.

Table 1 summarizes each attack category in detail. This follows the explanation in the previous text.

TABLE I
NSL-KDD DATASETS CHARACTERISTICS

NSL-KDD	Total Record	Normal Record	Dos Record	Probe Record	R2L Record	U2R Record
KDD_{Train+}	125973	67343	45927	11656	995	52
KDD_{Test+}	18793	9710	5741	1106	2199	37

B. Data Pre-processing

Data pre-processing aims to calculate data into a standard process so it can be properly routed to the next stage section. It also ensures that the machine learning algorithm can recognize the feature characteristic. To achieve the goal, the pre-processing process is divided into three sections: data normalization, outliers data analysis, and dimensional data transformation using one-hot-encoding.

1) Removing outlier: A value in the NSL-KDD is inconsistent, and Outliers frequently use this term to describe this problem. Before the normalization of the data step, it has an essential procedure. In addition, outliers may impact the proposed model of malicious detection, which could result in incorrect detection. We considered using Median Absolute Deviation Estimator (MADE), a technique whose working mechanism is represented in the following equation:

$$\text{MADE}=P*\text{med}\,(zfj - |\text{med}(zfj)|) \quad (1)$$

2) Data normalization: As part of the normalization process, the min-max method is used to calculate the Z_{fj} numerical attribute in the range of 0-1 with the following equation:

$$\tilde{z}_{fj} = \frac{z_{fj} - min(z_f)}{max(z_f) - min(z_f)} \quad (2)$$

3) One-hot-encoding: Protocol model, service, and flag are three special feature characteristic attacks that necessitate a specific method of handling (z2, z3, z4). To convert them into a numeric number, the one-hot-encoding method is required. Every categorical feature, in particular, was

demonstrated with a binary number. For example, protocol type is represented by three category attributes: udp, icmp, and tcp. The one-hot-encoding is in charge of the transformation into binary vector space, such as (1.0.0), (0.1.0), and (0.2.0). (0.0.1). The conversion process into a one-hot-encoding vector was also used for service and flag features with z3 and z4 symbol representation. The total number of feature attack characteristics in 41 features was computed into 122-dimensional features, which consisted of 84 dimensional features with binary class and 30 continuous values.

4) Dimensional reduction using SDAE: SDAE is a subclass of auto encoder (AE) neural network, in which the AE takes the input and transforms it into hidden layer representation using a deterministic mechanism, while the denoising autoencoder is in charge of extracting the input's missing representation layer [28]. This model aims to address the auto encoder problem, which is difficult to train in deep learning models in order to detect unsupervised learning processes that map feature inputs into middle process representations. According to the literature, some versions of autoencoders have been proposed and have demonstrated tremendous achievement in the field of computer science research [29]. Furthermore, a class denoising autoencoder can be stacked to compute a deep layer, as seen in high-level classes where it is known as stack denoising autoencoder. SDAE, in particular for the learning mechanism, uses regularization to address the optimization problem.

$$\min_{W_l, b_l} ||X_{corrupted\_input} - X_{output}||_F^2 + \lambda \sum_l (\|W_l\|_F^2 + \|b_l\|_2^2) \quad (3)$$

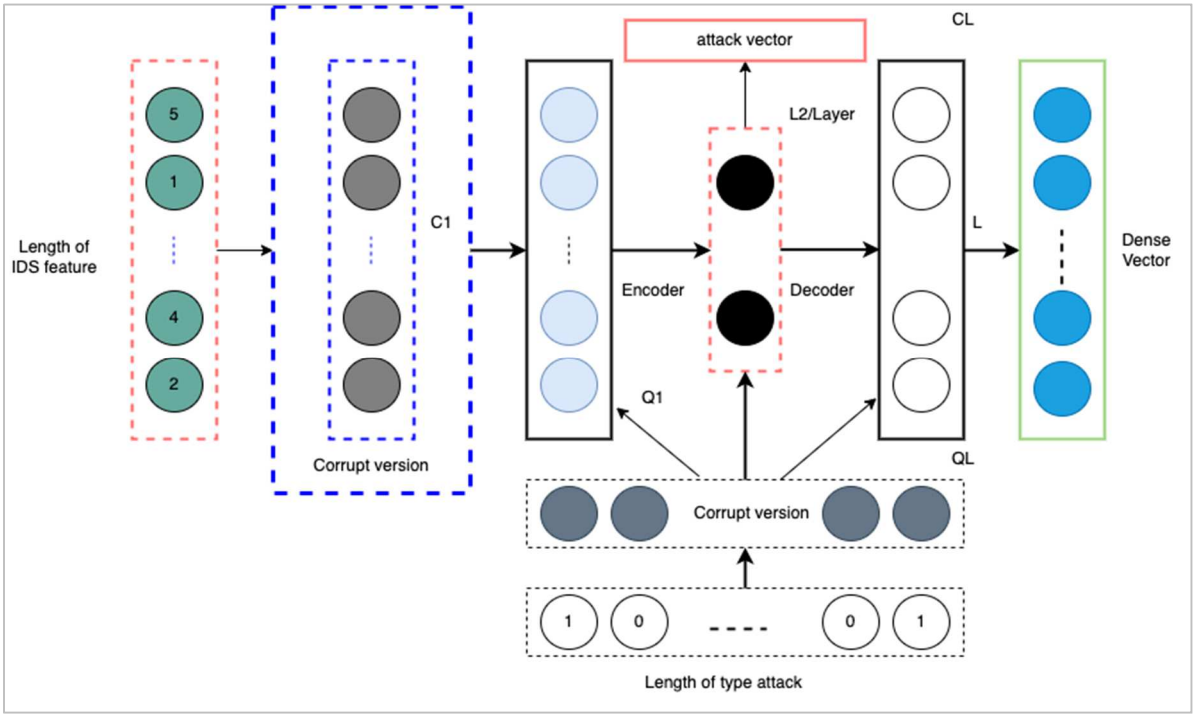

Fig. 3 SDAE Dimensional reduction framework

C. IDS Detection Classifier

This research considered incorporating four traditional classifier algorithms to observe the model's performance. The dimensional reduction using SDAE integrated into Naive Bayes, KNN, Decision Three, and SVM. The basic mechanism of the algorithm is explained below.

1) Naive Bayes: When dealing with binary (two classes) or multiclass classification problems, the Naive Bayes (NB) algorithm is the go-to choose. Binary or categorical input values make the technique easier to understand. Naive Bayes (also known as idiot Bayes) is a type of probability distribution that is simplified to make the calculation of the probabilities for each hypothesis tractable. To save time, rather than attempting to calculate the values of each attribute value $P(1)$, $P(2)$, and $P(3)|h$), it is assumed that they are conditionally independent given the target value and the values are calculated as $P(d1|h)$ * $P(d2|H)$ and so on.

2) K-nearest neighborhood (KNN): It is possible to use KNN, one of the simplest supervised machine learning algorithms, to predict the class of a particular data sample by considering "feature similarity." It calculates its distance from the other samples in the neighborhood to identify a sample. The parameter k in the KNN algorithm can affect the model's performance. At very small k values, the model may be subject to over-fitting problems. The sample instance may be incorrectly categorized if a large number of k values are selected [37], [38], [39].

3) Decision Tree: A Decision Tree (DS Tree) is a fundamental supervised machine learning algorithm that can be applied to both classification and regression problems on a given dataset (rules). Nodes, branches, and leaves make up the tree-like structure of the model. Each node is a feature or an attribute. Each leaf on the tree represents a possible outcome or classification, while the branch represents a rule or decision. To prevent over-fitting, the decision tree algorithm automatically selects the best features for creating

a tree and then performs pruning operations to remove irrelevant branches from the tree. These three decision tree models are the most widely used: CART, C4.5, and ID3 [40], [41].

4) Support Vector Machine (SVM): Using the SVM, a margin-based classification method, an optimum hyperplane is created that can effectively distinguish between the different classes as much as possible, following the principle of structural risk minimization [28]. As a result, SVM has a powerful generalization capability and is resistant to overfitting issues. Furthermore, SVM can deal with non-linear classification problems by selecting kernel functions to map the original feature space to some high-dimensional feature spaces with linearly separable instances.

D. Hybrid SDAE with Naive Bayes, KNN, Decision Tree, and SVM

Our study considers implementing SDAE and the popular traditional machine learning approach. It is a very important approach to observe the effectiveness level of several combinations between them. The schematic of the hybridization scheme can be seen in Figure 4 below. Our experiment consists of several evaluation processes, including multi-class and binary-class using confusion matrix, accuracy, recall, F1-measure, and precision. The multi-class experiment consists of 5 possibility conditions categories: normal, DoS, Probe, U2R, and R2L; while the binary class consists of 2 conditions: normal and anomaly.

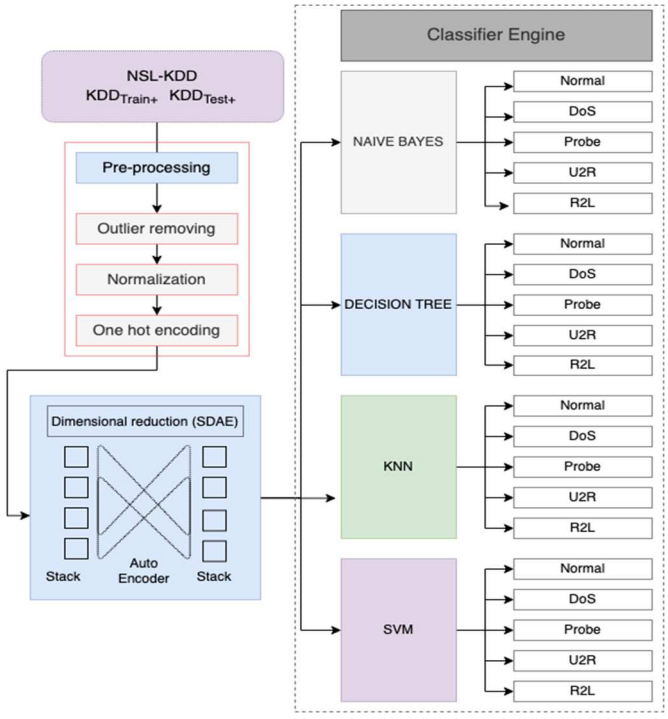

Fig. 4 Detail hybridization model and experiment scenario

We compared four traditional machine learning models including KNN, Naive Bayes, Decision Tree, and SVM. Then, they would be integrated into dimensional reduction based on SDAE respectively. SDAE is the enhancement of the Auto Encoder model. The advantage of variant Auto Encoder is that it is useful in feature extraction mechanisms. It is also a categorical modern deep machine learning. Our schematic training process divided the NSL-KDD into 30% and 70%. Most researchers in IDS detection have conducted this schematic training ratio.

E. Evaluation Metrics

For example, TP represents the true positive rate, which indicates the number of abnormal samples that tested positive (accurate detection). TN represents the true negative rate, indicating the number of normal samples tested negative (accurate detection). FP represents the false positive rate, representing how many abnormal samples tested positive (inaccurate detection). While FN represents the false-negative rate, which represents how many abnormal samples tested negative (accurate detection) (incorrect detection).

Accuracy is defined as the ratio of correctly classified samples to all samples in the testing set, expressed in percentage. Precision is the ratio of correctly classified samples to the total number of TP and FP samples in the testing set, expressed in percentage. The recall ratio is the ratio of the number of TP samples to the total number of TP and FN samples. When it comes to the time to compute the F1- score, it is calculated using the weighted average of precision and recall.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{4}$$

$$Precision = \frac{(TP)}{(TP+FP)} \tag{5}$$

$$Recall = \frac{(TP)}{(TP+FN)} \tag{6}$$

$$F1-score = 2 \times \frac{(Preciision \times Recall)}{(Precision + Recall)} \tag{7}$$

III. Results and Analysis

The result of dimensional reduction using SDAE can be seen in Fig. 5 below.

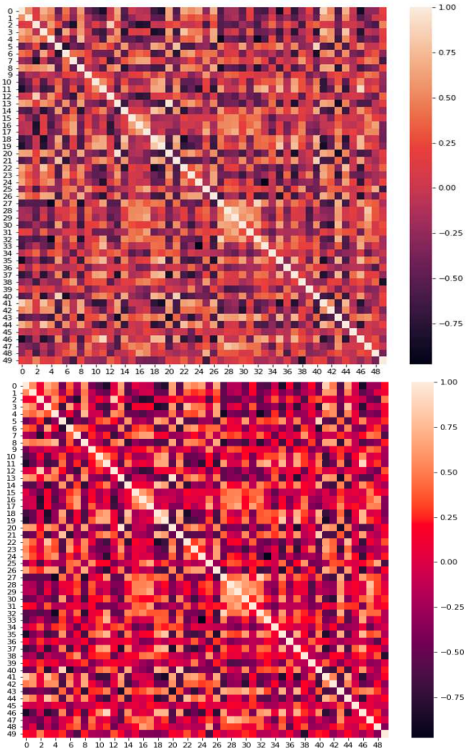

Fig. 5 SDAE training result of NSL-KDD

The dark colors represent values that are almost like the actual values, while the bright ones represent values that are very different from the actual values. Then, the output from dimensional reduction resulting from SDAE would be integrated into four machine learning categories. The evaluation metrics include accuracy, precision, recall, and F1 as shown in Table 2. The experiment of our model consisted of 2 classes which were multi-class and binary class, in which binary class only detected an anomaly and normal detection, while multi-class involved 5 categories condition including "Normal", "DoS", "Probe", "R2L", and "U2R".

As shown in Table II, the enhancement of dimensional reduction using SDAE succeeded to increase the effectiveness of traditional machine learning in IDS detection. The hybridization between SDAE and KNN model achieved an accuracy of 79.8% compared with KNN without SDAE, which only achieved 77.9%. The hybridization between SDAE and Naive Bayes also achieved better performance over the traditional Naive Bayes without SDAE with tremendous results in 80.5% compared to that of previous work results with 76.3%. Another successful model using a Decision Tree combined with SDAE achieved an accuracy of 83.4%, while the one without SDAE reached an accuracy of 82.9%. Our experiment report shows that SDAE and SVM achieved the best performance in 84.1%, whereas the traditional SVM only achieved an accuracy of 80%.

The multi-class training result shows that the combination of SDAE with 4 machines learning also reached better performance over traditional machine learning. The hybridization among SDAE and KNN reached an accuracy of 78.1%, while KNN without SDAE only achieved 75%. The novel hybridization between SDAE and Naive Bayes achieved better performance in 78.7% over traditional Naive Bayes which only reached 77.8%. Another hybridization model between Decision Tree and SDAE showed better performance in 82.8%. This achievement was 2% higher than the traditional Decision Tree, which only reached 80.1%. The hybridization reached the best achievement in our experiment between SDAE and SVM with an accuracy of 83.3%. It means that SDAE and SVM successfully increased the effectiveness level in IDS detection by more than 3% compared to the traditional SVM that only employed pre-processing process.

Our study also applied a confusion matrix to detect the effectiveness of our model. The confusion matrix was tried in each hybridization model and evaluated based on the multi-class and binary class classification approach. The binary class is shown in Fig. 6 to 13, while the multi-class classification can be seen in Fig. 14 to 21. Fig 6 to 13 demonstrated the involvement of SDAE, showing success in reducing misclass detection in every hybridization scenario, including SDAE with KNN, Naive Bayes, Decision Tree, and SVM. Hybridization between SDAE and KNN could increase accuracy detection by 81% from 79%. The combination between SDAE and Naive Bayes achieved 82.9% while traditional pre-processing and Naive Bayes only reached 81.7%. The combination between SDAE and Decision Tree showed better performance over previous work with KNN and Naive Bayes in which SDAE and Decision Tree reached 85.5% while the traditional Decision Tree and pre-processing only reached 82.1%. Meanwhile, the hybridization between SDAE and SVM has become the best performance with an accuracy of 86.2%. The traditional pre-processing and SVM reached 82.1%. The employment of SDAE proved more effective in every hybridization scenario in multi-class classification. This model is also effective in detecting 9341 normal network traffic with miss class detection in 946, and correct anomaly detection in 7274 with 1704 miss class detection.

TABLE II
COMPARISON RESULT ON BINARY CLASS MEASUREMENT

Evaluation result on binary classification				
	Accuracy	Precision	Recall	F1
SDAE & SVM	84.1%	85.6%	83.1%	84.3%
SDAE & DS Tree	83.4%	83.4%	79.6%	81.4%
SDAE & NB	80.5%	81.8%	78.9%	80.3%
SDAE & KNN	79.8%	81.1%	74.1%	77.4%
Pre-processing & SVM	80.7%	81.9%	78.7%	80.2%
Pre-processing & DS3	82.9%	83.3%	81.2%	82.2%
Pre-processing & NB	76.3%	77.6%	73.8%	75.6%
Pre-processing & KNN	77.9%	78.3%	75.8%	77.0%

TABLE III
COMPARISON RESULT ON MULTI-CLASS MEASUREMENT

Evaluation result on multi-classification				
	Accuracy	Precision	Recall	F1
SDAE & SVM	83.3%	85.1%	81.6%	83.3%
SDAE & DS Tree	82.8%	84.3%	80.8%	82.5%
SDAE & NB	78.7%	80.1%	76.1%	78.0%
SDAE & KNN	78.1%	79.7%	75.9%	77.7%
Pre-processing & SVM	80.0%	82.1%	77.8%	79.8%
Pre-processing & DS3	80.1%	82.7%	76.9%	79.6%
Pre-processing & NB	77.8%	79.1%	75.1%	77.0%
Pre-processing & KNN	75.6%	77.1%	72.3%	74.6%

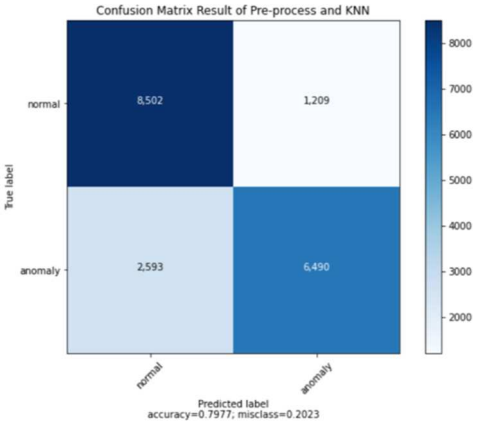

Fig. 6 Confusion matrix of pre-processing and KNN in the binary class

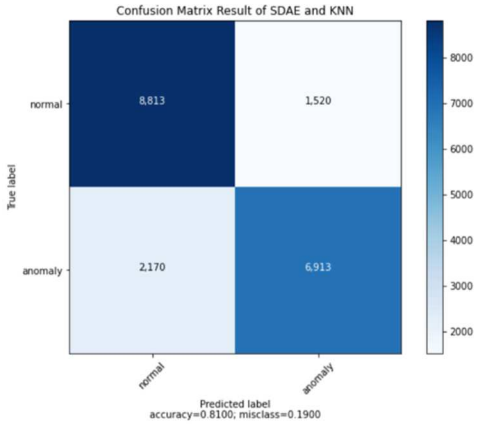

Fig. 7 Confusion matrix of SDAE and KNN in the binary class

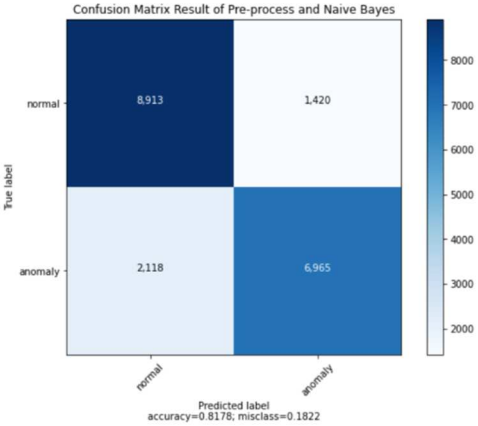

Fig. 8 Confusion matrix of pre-processing and Naive Bayes in the binary class

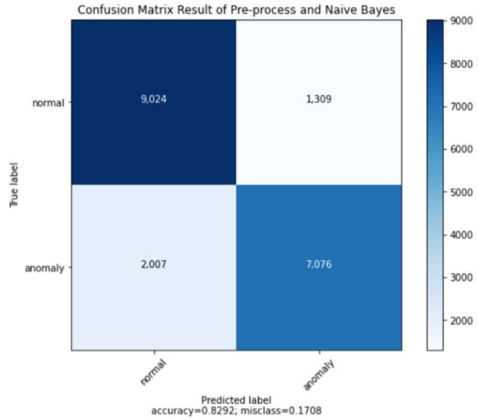

Fig. 9 Confusion matrix of SDAE and Naive Bayes in the binary class

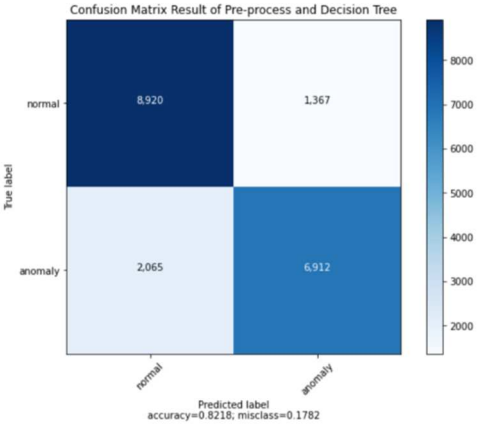

Fig. 10 Confusion matrix of Pre-processing and Decision Tree in the binary class

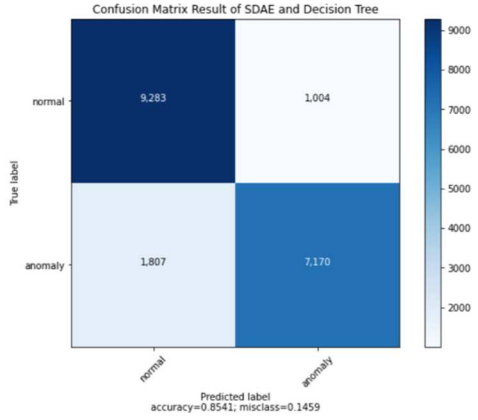

Fig. 11 Confusion matrix of SDAE and Decision Tree in the binary class

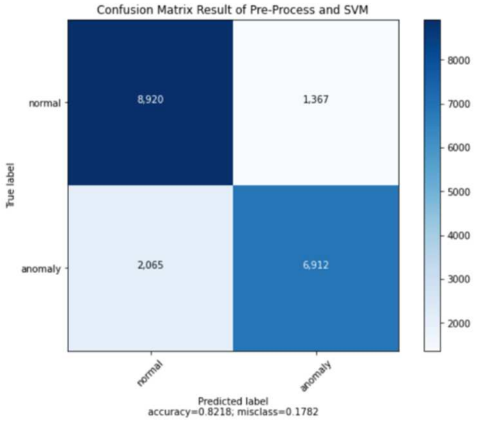

Fig. 12 Confusion matrix of Pre-processing and SVM in the binary class

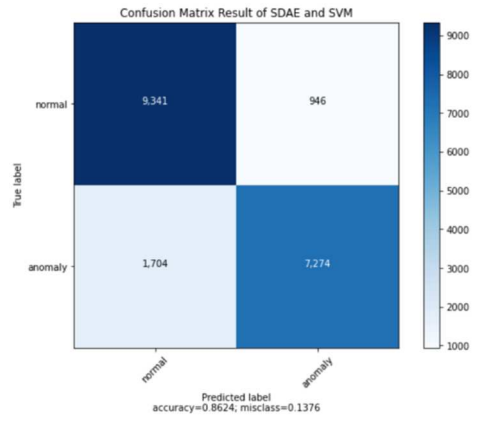

Fig. 13 Confusion matrix of SDAE and SVM in the binary class

The experiment report based on the confusion matrix on multi-class classification is shown in Fig. 14 to 21. Each figure shows that SDAE could reduce miss class detection. The involvement of SDAE supported KNN to enhance the accuracy level in confusion matrix evaluation by 74%, while the traditional KNN and pre-processing only reached 72%. The combination between SDAE and Naive Bayes also successfully increased performance in multi-class IDS detection in which this model achieved an accuracy of 79.9% compared to Naive Bayes and pre-processing, which reached an accuracy of 77.9%. The Decision Tree that applied SDAE also successfully reduced miss classification and increased accuracy in confusion matrix evaluation, which achieved 83.2%, whereas the Decision Tree without SDAE only reached 82%. Another hybridization model involving SDAE and SVM, evaluated using a confusion matrix, reached the best performance over the previous hybridization approach. SDAE-SVM could reduce miss classification, increase accuracy performance by 87%, and achieve an accuracy of 84% in pre-processing and SVM only.

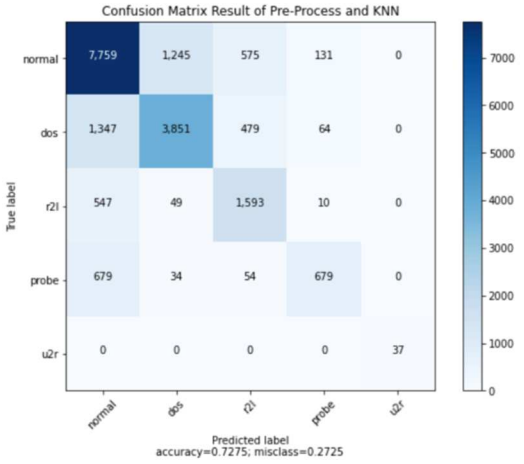

Fig. 14 Confusion matrix of Pre-processing and KNN in multi-class

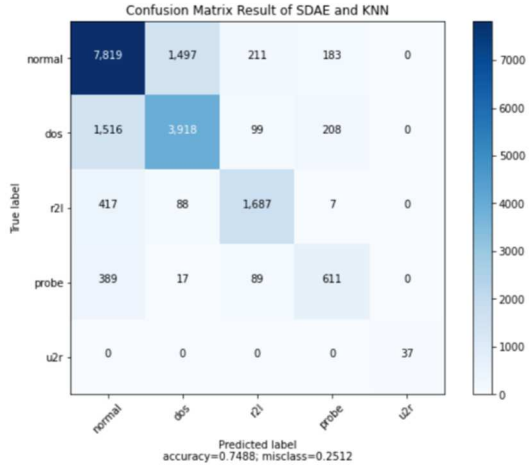

Fig. 15 Confusion matrix of SDAE and KNN in multi-class

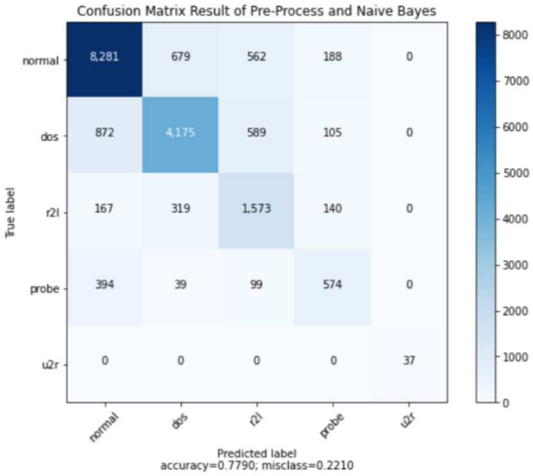

Fig. 16 Confusion matrix of Pre-processing and Naive Bayes in multi-class

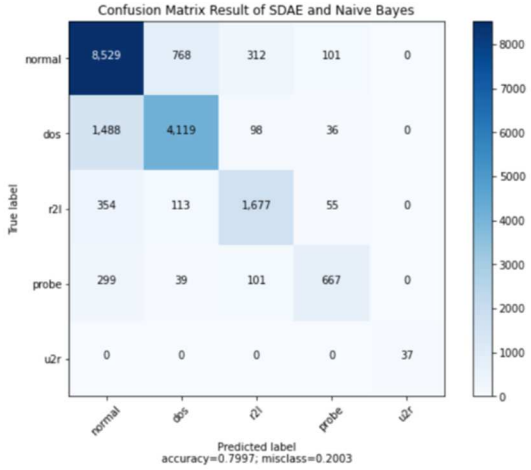

Fig. 17 Confusion matrix of SDAE and Naive Bayes in multi-class

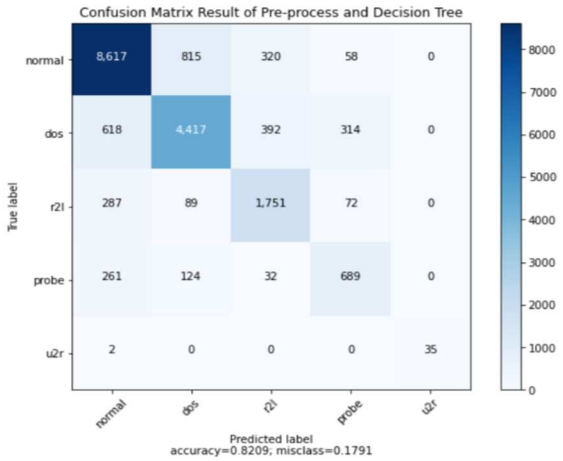

Fig. 18 Confusion matrix of Pre-processing and Decision Tree in multi-class

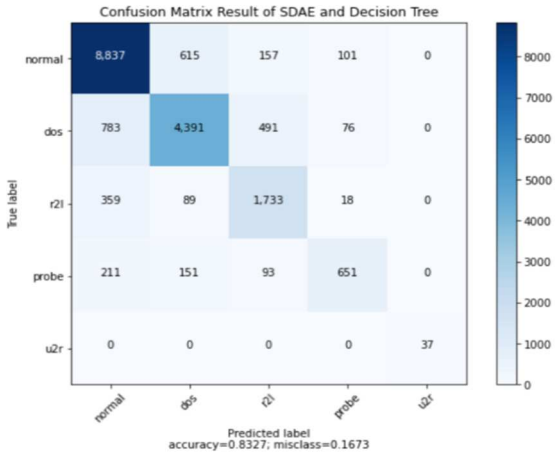

Fig. 19 Confusion matrix of SDAE and Decision Tree in multi-class

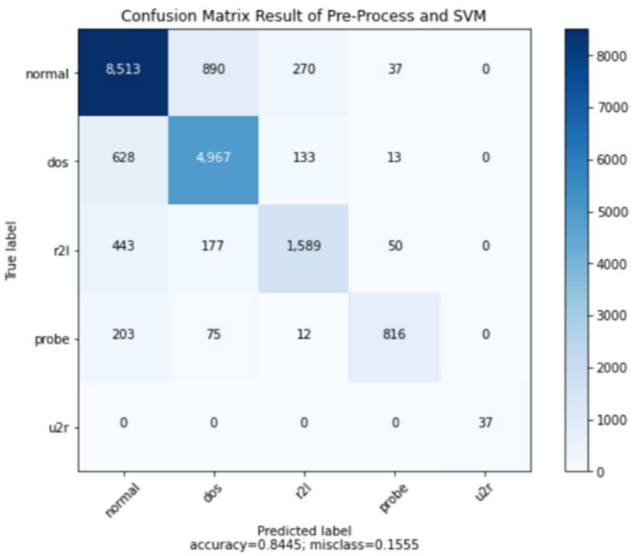

Fig. 20 Confusion matrix of Pre-processing and SVM in multi-class

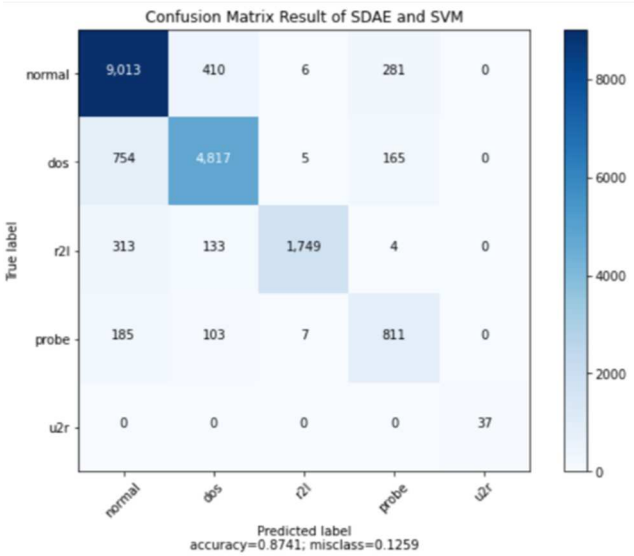

Fig. 21 Confusion matrix of SDAE and SVM in multi-class

The comparison results over the previous state-of-the-art have been conducted in this study. The competitor used several novel methods based on statistical and deep learning approaches, for instance, the hybridization of statistical models with machine learning, the combination between CNN and LSTM, LSTM and Mutual information, and LSTM and PCA. The comparison is shown in Table 3.

TABLE IV
COMPARISON RESULT OVER STATE-OF-THE-ART

No	Model	Accuracy
1	SDAE & SVM (our model)	**84.1%**
2	SDAE & Decision Tree (our model)	83.4%
3	SDAE & Naive Bayes (our model)	80.5%
4	SDAE & KNN (our model)	79.8%
5	CNN & LSTM (BAT) [42]	**84.25%**
6	Statistic & ML [43]	83.65%
7	LSTM & PCI [35]	82.4%
8	LSTM & MI [35]	81.8%

IV. CONCLUSION

This present study considers enhancing dimensional reduction using a variant of auto encoder based on SDAE. It is found that this model is useful for improving the traditional machine learning work. SDAE is also suitable for reducing miss classification in traditional machine learning such as KNN, Naive Bayes, Decision Tree, and SVM. SDAE and SVM achieved the best combination in our experiment compared to the other models, such as Decision Tree (the second-best achievement), Naive Bayes, and KNN.

SDAE also successfully increased the effectiveness of classification mechanisms in machine learning, especially in IDS detection, even when compared to modern machine learning approaches such as deep learning based on CNN and LSTM in binary and multi-class classification methods.

There are some challenges in future research in that SDAE can be integrated with modern deep learning approaches such as MLP, LSTM, CNN, and GAN to reduce miss class prediction and increase the correct value prediction. Our model that is developed using traditional machine learning is highly possible to be improved with an ensemble learning approach.

ACKNOWLEDGMENT

This study is supported by Universitas Amikom Yogyakarta, Indonesia.

REFERENCES

[1] B. Zarpelão, R. Miani, … C. K.-J. of N. and, and undefined 2017, "A survey of intrusion detection in Internet of Things," *Elsevier*, 2017, doi: 10.1016/j.jnca.2017.02.009.

[2] K. N. L. biswanath Mukherjee, L. Todd Heberlein, "Network Intrusion Detection," *IEEE Netw.*, 1994.

[3] S. Wagh, A. ali shah, S. Kishor Wagh, V. K. Pachghare, and S. R. Kolhe, "Survey on Intrusion Detection System using Machine Learning Techniques," *Int. J. Comput. Appl.*, vol. 78, no. 16, pp. 975–8887, 2013.

[4] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches Emulated Monitoring Systems View project Deep Learning View project Survey on SDN based network intrusion detection system using machine learning approaches," doi: 10.1007/s12083-017-0630-0.

[5] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00379-6.

[6] F. E. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "IDS-attention: an efficient algorithm for intrusion detection systems using attention mechanism," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00544-5.

[7] H. Zhang, "Design of intrusion detection system based on a new pattern matching algorithm," *Proc. - 2009 Int. Conf. Comput. Eng. Technol. ICCET 2009*, vol. 1, pp. 545–548, 2009, doi: 10.1109/ICCET.2009.244.

[8] C. Yin, "An Improved BM Pattern Matching Algorithm in Intrusion Detection System," *Appl. Mech. Mater.*, vol. 148–149, pp. 1145–1148, 2012, doi: 10.4028/www.scientific.net/AMM.148-149.1145.

[9] D. E. Denning, "An Intrusion-Detection Model," *IEEE Trans. Softw. Eng.*, vol. 13, no. 2, pp. 222–232, 1987.

[10] M. Pervez, D. F.-T. 8th I. C. on, and undefined 2014, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," *ieeexplore.ieee.org*, 2015, doi: 10.1109/SKIMA.2014.7083539.

[11] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 38, no. 5, pp. 649–659, 2008, doi: 10.1109/TSMCC.2008.923876.

[12] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," *Int. Conf. Signal Process. Commun. Eng. Syst. - Proc.*

SPACES 2015, Assoc. with IEEE, pp. 92–96, Mar. 2015, doi: 10.1109/SPACES.2015.7058223.

[13] B. Ingre, A. Yadav, and A. K. Soni, "Decision Tree Based Intrusion Detection System for NSL-KDD Dataset," *Smart Innov. Syst. Technol.*, vol. 84, pp. 207–218, 2017, doi: 10.1007/978-3-319-63645-0_23.

[14] N. Rusk, "Deep learning," *Nat. Methods*, vol. 13, no. 1, p. 35, 2015, doi: 10.1038/nmeth.3707.

[15] Hanafi, A. Pranolo, and Y. Mao, "Cae-covidx: Automatic covid-19 disease detection based on x-ray images using enhanced deep convolutional and autoencoder," *Int. J. Adv. Intell. Informatics*, vol. 7, no. 1, pp. 49–62, 2021, doi: 10.26555/ijain.v7i1.577.

[16] Hanafi and B. M. Aboobaider, "Word Sequential Using Deep LSTM and Matrix Factorization to Handle Rating Sparse Data for E-Commerce Recommender System," *Comput. Intell. Neurosci.*, vol. 2021, no. 1, 2021, doi: https://doi.org/10.1155/2021/8751173 Research.

[17] Hanafi, E. Pujastuti, A. Laksito, A. Arfriandi, R. Hardi, and R. Perwira, "Handling Sparse Rating Matrix for E-commerce Recommender System Using Hybrid Deep Learning Based on LSTM , SDAE and Latent Factor," vol. 15, no. 2, pp. 379–393, 2022, doi: 10.22266/ijies2022.0430.35.

[18] Hanafi, N. Suryana, and A. S. B. H. Basari, "Recommender System Based Tensor Candecomp Parafact Algorithm-ALS to Handle Sparse Data In Food Commerce Information Services," *IJSSST*, pp. 1–9, 2019, doi: 10.5013/IJSSST.a.19.06.60.

[19] Hanafi, N. Suryana, and A. S. B. H. Basari, "Convolutional-NN and word embedding for making an effective product recommendation based on enhanced contextual understanding of a product review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, 2019, doi: 10.18517/ijaseit.9.3.8843.

[20] Hanafi, N. Suryana, and A. S. B. H. Basari, "Paper Survey and Example of Collaborative Filtering Implementation in Recommender System," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 16, 2017.

[21] Hanafi, R. Widyawati, and A. S. Widowati, "Effect of service quality and online servicescape toward customer satisfaction and loyalty mediated by perceived value," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 704, no. 1, 2021, doi: 10.1088/1755-1315/704/1/012011.

[22] Hanafi, N. Suryana, and A. S. H. Basari, "Generate Contextual Insight of Product Review Using Deep LSTM and Word Embedding," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, 2020, doi: 10.1088/1742-6596/1577/1/012006.

[23] Hanafi, N. Suryana, and A. S. H. Basari, "Deep Contextual of Document Using Deep LSTM Meet Matrix Factorization to Handle Sparse Data: Proposed Model," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, 2020, doi: 10.1088/1742-6596/1577/1/012002.

[24] Hanafi, N. Suryana, and A. S. H. Basari, "Involve Convolutional-NN to Generate Item Latent Factor Consider Product Genre to Increase Robustness in Product Sparse Data for E-commerce Recommendation," *J. Phys. Conf. Ser.*, vol. 1201, no. 1, 2019, doi: 10.1088/1742-6596/1201/1/012004.

[25] Hanafi, N. Suryana, and A. Samad, "Dynamic convolutional neural network for eliminating item sparse data on recommender system," *IJAIN*, vol. 4, no. 3, pp. 226–237, 2018.

[26] A. Javaid, Q. Niyaz, W. Sun, M. A.-E. E. T. on, and undefined 2016, "A deep learning approach for network intrusion detection system," *eprints.eudl.eu*, 2016, doi: 10.4108/eai.3-12-2015.2262516.

[27] G. Zhao, C. Zhang, L. Z.-2017 I. International, and undefined 2017, "Intrusion detection using deep belief network and probabilistic neural network," *ieeexplore.ieee.org*, 2017, doi: 10.1109/CSE-EUC.2017.119.

[28] F. Qu, J. Zhang, Z. Shao, S. Q.-P. of the 2017 V. international, and undefined 2017, "An intrusion detection model based on deep belief network," *dl.acm.org*, pp. 97–101, Dec. 2017, doi: 10.1145/3171592.3171598.

[29] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *National Aerospace and Electronics Conference (NAECON)*, 2015, pp. 333–344, doi: 10.1109/NAECON.2015.7443094.

[30] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," *2017 IEEE Int. Conf. Big Data Smart Comput. BigComp 2017*, pp. 313–316, Mar. 2017, doi: 10.1109/BIGCOMP.2017.7881684.

[31] K. Wu, Z. Chen, and W. Li, "A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018, doi: 10.1109/ACCESS.2018.2868993.

[32] K. Hara and K. Shiomoto, "Intrusion Detection System using Semi-Supervised Learning with Adversarial Auto-encoder," *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. 2020 Manag. Age Softwarization Artif. Intell. NOMS 2020*, 2020, doi: 10.1109/NOMS47738.2020.9110343.

[33] S. Hochreiter and J. Urgen Schmidhuber, "Lstm," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[34] Hanafi and andi sunyoto, "Enhance Intrusion Detection (IDS) System Using Deep SDAE to Increase Effectiveness of Dimensional Reduction in Machine Learning and Deep Learning," vol. 15, no. 4, pp. 125–141, Jun. 2022.

[35] F. E. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00448-4.

[36] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. July, 2009, doi: 10.1109/CISDA.2009.5356528.

[37] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *J. Electr. Comput. Eng.*, vol. 2014, no. 1, 2014, doi: 10.1155/2014/240217.

[38] R. Taguelmimt and R. Beghdad, "DS-kNN: An intrusion detection system based on a distance sum-based K-nearest neighbors," *Int. J. Inf. Secur. Priv.*, vol. 15, no. 2, pp. 131–144, 2021, doi: 10.4018/IJISP.2021040107.

[39] S. Choi, "Combined kNN classification and hierarchical similarity hash for fast malware detection," *Appl. Sci.*, vol. 10, no. 15, pp. 1–16, 2020, doi: 10.3390/app10155173.

[40] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "RDTIDS: Rules and decision tree-based intrusion detection system for internet-of-things networks," *Futur. Internet*, vol. 12, no. 3, pp. 1–14, 2020, doi: 10.3390/fi12030044.

[41] K. Rai, M. S. Devi, and A. Guleria, "Decision Tree Based Algorithm for Intrusion Detection," *Int. J. Adv. Netw. Appl.*, vol. 07, no. 04, pp. 2828–2834, 2016.

[42] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[43] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020, doi: 10.1016/j.neucom.2019.11.016.