Received: February 9, 2022. Revised: April 23, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhance Intrusion Detection (IDS) System Using Deep SDAE to Increase Effectiveness of Dimensional Reduction in Machine Learning and Deep Learning

Andi Sunyoto[1] **Hanafi[1]***

[1]*Department of computer science, Universitas Amikom Yogyakarta,*
Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta, 55283 Indonesia
* Corresponding author's Email: hanafi@amikom.ac.id

Abstract: The intrusion detection system (IDS) is very essential tools to detect malicious network. IDS is a hardware or software approach to observe the internet for malicious attacks. It has ability to screening an internet or network that possibility dangerous activity or security threats. IDS application responsible to defend network territory in accordance with the network-based intrusion detection system (NIDS) or host-based intrusion detection system (HIDS). Using known normal network activity signatures, IDS applications perform tasks by comparing them to known attack activity signatures. In this study, a dimensional reduction and feature selection mechanism known as the stack denoising auto encoder (SDAE) was found to be effective in increasing the effectiveness of naive bayes, KNN, decision tree, and SVM classification algorithms. The researchers evaluated the performance using evaluation metrics such as a confusion matrix, accuracy, recall, and the F1-score, among other measures of success. When compared to the results of previous studies in the IDS field, our model using statistical pre-processing, dimensional reduction based on SDAE success to increase the effectiveness of KNN, naive bayes, decision tree, SVM and deep learning using LSTM. We applied our experiment in the NSL-KDD Dataset. According to evaluation metrics using confusion matrix, accuracy, recall, and f1 that the effectiveness of our model achieve more than 2% over several previous work without statistical pre-processing and dimensional reduction based on SDAE. Furthermore, the use of statistical approaches and SDAE improved the accuracy of traditional machine learning and modern deep learning based on LSTM. Aims to improve the effectiveness of IDS detection in the future, it may be possible to integrate SDAE with another deep learning model such as MLP, CNN, attention, and GAN.

Keywords: IDS detection, SDAE, Naive bayes, Decision tree, SVM, LSTM.

1. Introduction

Over the last decade, the number of internet's user growth significantly. As a result of recent technological advancements, particularly those relating to the internet, communication and networking, a massive amount of data has been generated from a variety of sources such as the industrial sector, online shopping portals, messenger services, social media, and health-care providers. Big data is representing to describe a massive amount of data that has four characteristics: high veracity, high velocity, high variety, and high value, to name a few. As a result of the widespread use of big data also automatically increase the number of cyberattacks. More than 26 billion devices were connected to the internet in 2019, according to statistics. In addition, it contributes to the expansion of malicious activity on the internet in general. Adoption of IDS malicious network detection has evolved into an essential application for enhance the security of computer networks and computer systems [1, 2].

In order to improve IDS, a large number of experts, researchers, and academicians have used conventional machine learning mechanism as the most popular machine learning algorithm such as neural networks (NN), support vector machines (SVM), K nearest neighbours (KNN), Decision tree (DS3), multilayer perceptron (MLP), and auto encoder (AE). The use of conventional shallow learning frameworks (one feedforward network) to

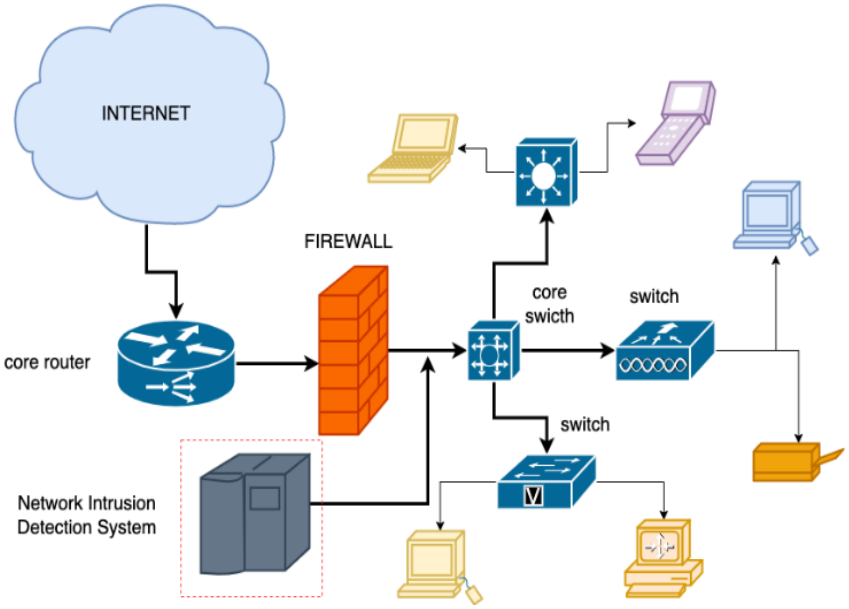


Figure. 1 The illustration of IDS attack network detection

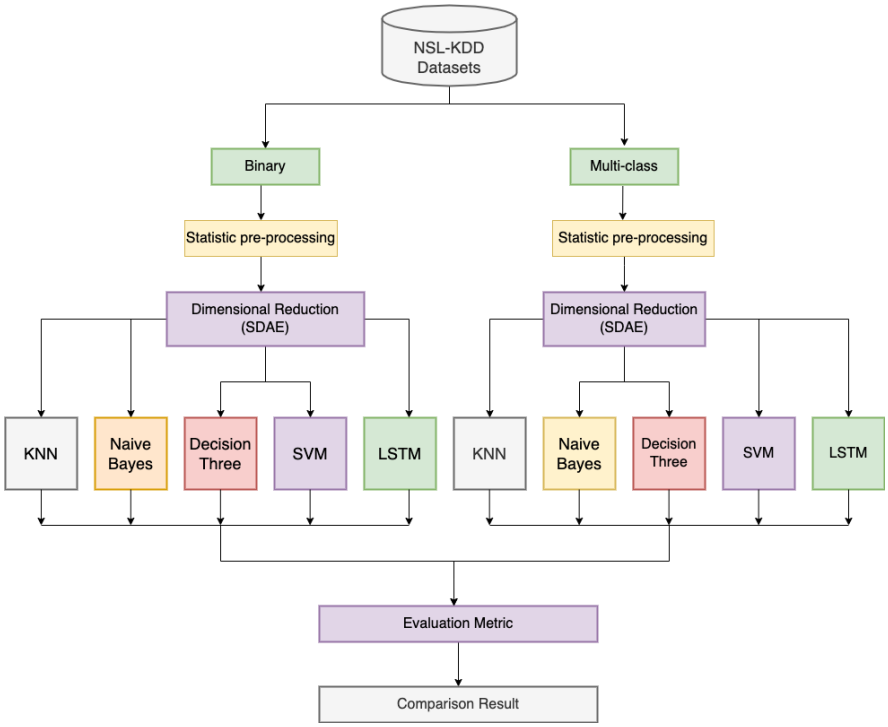


Figure. 2 Scenario of IDS network attack detection experiment

solve the auto detection problem for big data is ineffective because of the large amount of data involved. The failure to detect activity attacks, capture attack information accurately, and resolve noise in massive datasets is a recurring problem for them [3, 4]. It has become increasingly popular in recent years to use deep learning framework, for instance a convolutional neural network (CNN), a gated recurrent unit (GRU), and a long short-term memory (LSTM) in response to the aforementioned issue [5]. The illustration of IDS network attack detection can be seen in Fig. 1.

In addition, the number of attributes that IDS must look for in the data that they get from the internet is always huge, even in small-capacity networks. Indeed, a lot of raw data is useless and noisy. As a result, the classifier's performance suffers because it has features that aren't good enough. The principal component analysis (PCA), mutual information (MI), chi-square, and UMAP are all multidimensional reduction frameworks that can help

you get rid of a lot of information [6]. Our experiment used SDAE to make enhancement of dimensional reduction. The experiment scenario can be seen on Fig. 2.

In this study, our algorithm model involves a statistical model for pre-processing, dimensional reduction model based on SDAE, and hybridization with popular traditional machine learning model including KNN, naive bayes, Decision tree and SVM. The second contribution, we consider to hybrid between statistical approach for pre-processing mechanism, dimensional reduction using SDAE and modern deep learning lead to sequential aspect mechanism based on LSTM.

2. Related work

According to a large number of previous studies, the intrusion detection model employs three primary methods: deep learning, conventional machine learning, and pattern similarity. Deep learning has surpassed all other methods in popularity over the last few years. Initially, pattern similarity models were used primarily to detect intrusions. The majority of them employ patterns that are similar to their primary core learning algorithm, and they do so via attribute similarity [7, 8]. The majority of frameworks have already been implemented 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 all examples of traditional models used to create an intrusion detection system. Following the results of the experiments, it was determined that an algorithm worked well for speeding up and reducing the time required to perform pattern similarity calculations. However, the traditional model of pattern similarity has a significant flaw. They are unable to comprehend how intrusion detection works. The development of a low-cost algorithm capable of reducing the time required and the cost of false positives has become the primary objective of this study.

Denning [9] was the first person to made that IDS machines learning, and his study used a multi-algorithm model to look for intrusion detection activity. An expert thinks that the model made a pattern of several features by hand. In the beginning, a modern machine learning model based on the SVM was made [10]. The experiment set up KDD99 datasets, and it came up with 3 features with an accuracy of 91%, 36 features with an accuracy of 99%, and 41 features with an accuracy of 99%.

A study that used traditional machine learning techniques as well as KNN was successful in improving an early model. A K-mean clustering algorithm and a KNN classifier were used in this model [11]. CANN is the name given to the state-of-the-art IDS intelligence machine for malicious detection that evolved from this model. Another study [11, 12] proposes the use of a traditional classifier in conjunction with random forest to improve CANN performance. With an accuracy of 94.7%, the hybrid model, which used random forest as its core classifier machine, outperformed its competitors. The use of an artificial neural network (ANN) to improve the performance of a random forest (RF) has been proposed [13]. Application of the ANN model to NSL-KDD resulted in greater than 81 percent accuracy and 79% classification for malicious detection and network attack classification, as well as for network attack classification. There has been a proposal for a Decision tree (DT) intrusion detection model based on NSL-KDD [14]. It was determined that DT was effective in the IDS detection classification task as a result of the experiment's findings, which were published online. According to the explanation provided above, the enhancement of traditional machine learning results in astounding effectiveness in the detection of IDS threats. However, the majority of them necessitated extensive pre-processing on a large scale as well as complex attribute extraction. When employing a machine learning classification method, it is impossible to deal with large amounts of intrusion information.

In the early decade, deep learning, a new type of neural network with an extremely complex network structure was firstly introduced. At the time, deep learning achieved phenomenal performance in the image processing classification task. Additionally, deep learning has established itself as the industry standard for a variety of computer science-related problems, including image processing [15], speech recognition, recommender system using contextual document based on CNN [16-18], LSTM [19, 20], SDAE [21], Word embedding and LSTM [22].

According to Ref [23], a deep learning model based on auto encoder was proposed. They investigated the self-taught learning (STL) model with the help of NSL-KDD. It is composed of two fundamental process classifications, which are described below. It is necessary to train a dataset with unlabelled data before moving on to the next step in the compact attribute representation process. The second step is to train the learning representation features with labelled data and then to implement the classification of IDS tasks using the learning representation features trained. STL was used in three different levels of the experiment: two, five, and twenty-three classes. Results showed that STL had an

accuracy of 88.39%, and that the 5-class classification had an accuracy of 79.10%, with the 5-class classification having the highest accuracy.

Combining deep belief networks (DBNs) and probabilistic neural networks resulted in a deep learning model [24]. DBN is in charge of converting low-dimensional representations to non-linear representations while retaining critical raw data characteristics. They use particle swarm optimization to optimize hidden layer learning. Additionally, probabilistic neural network (PNN) detection of IDS makes use of final classification techniques. As demonstrated in their experiment, DBN-PNN achieved a 93.25% accuracy rate. Furthermore, DBN-PNN outperformed previous research that combined principal component analysis (PCA) mechanism to enhance dimensional reduction and combining with probabilistic neural networks (PNN).

Another study proposed a deep belief network (DBN) based deep learning model for the IDS task [25, 26]. Two critical processes are incorporated into this model: They did two factors: 1) they learned layer by layer using a restricted Boltzmann machine (RBM), and 2) they deduced the hidden layer vector from the visible layer vector. The representation of the hidden layer is the vector manifest for the subsequent layer. Both processes combine backpropagation networks generated by the final RBM method and use the final RBM output vector as an input vector. The DBM model achieves a 95.25% measurement accuracy. This results in an 89.07% performance advantage over backpropagation and a 91.36% performance advantage over SVM.

DNN is an abbreviation for deep neural network, which is considered suitable for use in intrusion detection systems [27]. It is a representation of an auto encoder that has four hidden layers and one hundred hidden units, as represented by the DNN algorithm. When they want to activate the hidden layer, they use rectified linear units (ReLU). ReLU categorizes activation functions that are not linear in their behaviour. The purpose of this activation function is to improve the algorithm's performance when performing complex classification tasks, such as identifying patterns in data. For the purpose of reaching the stochastic optimizer, this study made use of the adaptive moment mechanism. The experiment demonstrated that DNN was capable of measuring with 99% accuracy, as demonstrated in the experiment.

A novel model for detecting IDS networks has been proposed using convolutional neural networks (CNN) [28]. The CNN model is well-suited for a wide variety of image processing problems. The author made the assumption in this IDS detection case that the image processing problem is comparable to the IDS problem in terms of data vector dimension. CNNs are a subclass of feedforward neural networks that use convolutional processes to reduce large amounts of multidimensional data to representative vectors. This work, which makes use of a CNN model, asserts that the model was successful in improving the imbalanced dataset and that it not only decreased the false alarm rate but also improved the class's accuracy even when the sample size was small. As stated in their experiment report, CNN achieves a 79.48% accuracy in KDD-NSL. It outperforms a number of previously proposed conventional machine learning techniques.

A novel IDS detection model [29], was tested using GAN (generative adversarial network) and AE techniques. In their application of a semi-supervised model applied on NSL-KDD dataset, they were able to reduce the amount of time and effort required to manually tag the labelled data while simultaneously increasing the effectiveness of IDS malicious detection without labelled data. Using GANs and AEs to improve IDS detection on NSL-KDD datasets was a successful experiment report, even with only 0.1% of the datasets containing labelled data being used as training data.

It is a subclass of feedforward neural networks with sequential aspect mechanisms that is known as the long short-term memory (LSTM). It is a recurrent neural network enhancement that is being discussed. This year, LSTM is being considered as a possible model for an IDS network, such as the so-called DL-IDS, which is currently under development. According to the results of an experiment on hybrid PCA/LSTM [30], the DL-IDS has an accuracy rate of 98.67%. PCA is in charge of reducing the size of raw data attacks, while LSTM is in charge of categorizing network attacks. They claim that PCA-LSTM achieves 99.45% accuracy in binary class and 99.39% accuracy in multiclass classification when used in binary class. By reducing the number of dimensions in the PCA model, it was possible to improve the performance of the LSTM. In their research, they also proposed the concepts of mutual information (MI) and LSTM. It has a binary class accuracy of 96.24% and a multi-class classification accuracy of 95.56%, respectively.

3. Material and method

In this research, author consider to applying statistical approach to enhance pre-processing method before employ within SDAE to advance dimensional reduction, and integrated into several hybridization scenarios with traditional machine

Table 1. 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

learning based on KNN, naive bayes, Decision tree, SVM and deep learning approach based on sequential mechanism that famous called LSTM. The detail explanation of material and methodology on section below.

3.1 NSL-KDD datasets explanation

NSL-KDD is an advance variant of the KDD99 datasets. This dataset is very popular to compare of the effectiveness in many IDS network application and research. NSL-KDD enhances several drawbacks from the genuine KDD99 datasets in repetition and replication point of view. This condition contains in testing and training data record. It will impact in bias classification engine toward frequent sample.

NSL-KDD develop for free access in public society. It was made by Canadian cyber security institute [31]. This dataset contain two essential categories include training and testing that configured as KDD_{Train+} and KDD_{Test+} respectively. This dataset consists of 125973 training information and 22544 testing information. Started with the KDD_{Test+} introduced extra 17 attack type where this is not included into KDD_{Train+}. Aims to make classification output IDS detection fair enough, many researcher and academia deleted 3751 attack type that was unnecessary category. Finally, the the KDD_{Test+} actually contain 22544 - 3751 = 18793. According to Table 1 can be seen the specification of the KDD_{Train+} and KDD_{Test+}. NSL-KDD also include characteristic with z_f (f=1,2,3,4,5,..41) feature where it is contain 3 symbol and 38 continuous attributes categories. This dataset also consist to 4 attack classification categories as follows:

1. **Denial of service (DoS):** A DoS attack occurs when someone attempts to block access to a network service, server, or other service by flooding the internet with traffic. Someone else can slow or shut down a server or network service during a DoS attack.
2. **Root to local (R2L):** R2L attacks send bogus remote packets to a server or computer system in order to gain access to the server or computer system without permission.
3. **User to root (U2R):** U2R is a set of attacks aimed at gaining access to a computer's "root" area. In this case, the hacker discovers the flaw in the system and logs in as a regular user.
4. **Probe:** Probe is a type of attack that can gather information about networks and security management systems without being controlled by anyone.

Table 1 below is the complete description of NSL-KDD characteristic include number of record each attack category.

3.2 Data pre-processing with statistical scenario

The objective of pre-processing of NSL-KDD datasets is to transform the raw data using standard process. So, it can be work properly and adequate to next section process. It also aims to make sure that the attribute of attack characteristic can be recognized by the machine learning mechanism. Aim to reach the objective, the pre-processing stage consist to four sessions. The complete explanation of each statistical pre-processing as follow:

- **Deleting outlier**: A value in the NSL-KDD is inconsistent, so we need to remove it. This problem is frequently referred to as the "outlier problem" by these individuals. There is a necessary procedure that comes before the normalization of the data. The proposed model for detecting malicious activity may be affected by outliers, which could lead to incorrect detection. Median absolute deviation estimator (MADE), a technique that uses the following calculation as its working mechanism:

$$MADE=P\times med(z_{fj}\text{-}|med(z_{fj})|)$$

- **Data normalization process:** The min-max method is applied to compute the z_{fj} numerical attribute in the range of 0-1 as part of the normalization process, using the following calculation as:

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

One-hot-encoding process: Attacks that target the protocol model, service, or flag all fall under the umbrella term of one-hot encoding (z_1, z_2, z_3,

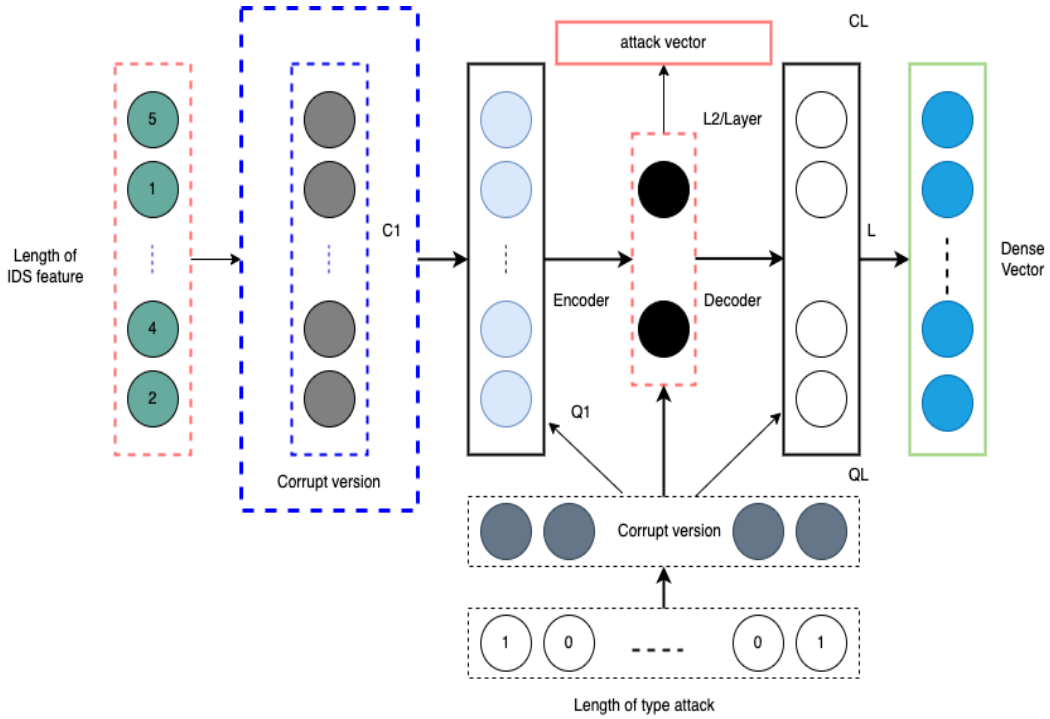


Figure. 3 SDAE framework for dimensional reduction

z_4). The one-hot-encoding method is required to transform them into a numeric value. A binary number was applied to demonstrate each feature's categorization. For instance, udp, icmp, and tcp are 3 category attributes that represent protocol type. It is the responsibility of the one-hot-encoding to transform the binary vector space into values such as (1.0.0), (0.1.0), and (0.2.0). (0.0.1). Service and flag features with z_3 and z_4 symbol representation were also converted into a one-hot-encoding vector. For each feature, 122 dimensions (84 binaries and 30 continuous) were computed to represent the total number of attack characteristics in 41 features.

3.3 Dimensional reduction using SDAE

Succinctly defined as a subclass of auto encoder neural networks, in which the auto encoder (AE) neural network takes the input and transforms it into hidden layer representation using a deterministic mechanism, while the denoising auto encoder (DAE) neural network is responsible for extracting the input's missing representation layer. Aims of this model include resolving the auto encoder problem, which is notoriously difficult to train in deep learning models, and detecting unsupervised learning processes that map feature inputs into middle process representations. A number of auto encoders have been proposed in the literature, and some of these versions have demonstrated tremendous success in the field of computer science research [32, 21]. It is also possible to stack multiple instances of a class denoising auto encoder in order to compute a deep layer, as seen in high-level classes, where this is known as a stack denoising auto encoder. Regularization is used to address the optimization problem in SDAE, and this is particularly true for the learning mechanism.

$$\min_{W_l,b_l} ||X_{corrupted_{input}} - X_{output}||_F^2 + \lambda \sum_l (\|W_l\|_F^2 + \|b_l\|_2^2) \quad (1)$$

3.4 Classifier engine for IDS attack network detection

This research considered to incorporating four traditional classifier algorithms to observe the hybridization model ability. The pre-processing used statistical approach and the dimensional reduction using SDAE integrated into naive bayes, KNN, Decision tree, and SVM and the last experiment, we tried to integrating statistical pre-processing and SDAE into variant of deep learning machine called LSTM. The basic mechanism of the algorithm is explained below.

3.4.1. Naive bayes

A Naive bayes classifier is a straightforward probabilistic classifier that is based on Bayes'

theorem and strict (naive) independence calculation. A Naive bayes classifier makes the prediction that the presence (or absence) of a particular feature within a class is unrelated to the presence (or absence) of any other feature within the class. Naive bayes classifiers can be trained very efficiently in a supervised learning setting, depending on the precise nature of the probability model. The Bayes theorem can be stated as follows:

$$P(H|X) = P(X|H)\,P(H) / P(X)\,P(H)$$

Let X denote the data record and H denote some hypothesis representing X as a member of a particular class C. We want to determine $P(H|X)$ for classification purposes, which is the probability that the hypothesis H holds true given an observed data record X. $P(H|X)$ denotes the posterior probability of H in the presence of X. $P(H)$ on the other hand is the prior probability. The posterior probability $P(H|X)$ is based on more information, such as prior knowledge, than the independent prior probability $P(H)$. Similarly, $P(X|H)$ denotes the posterior probability of X given H. The Bayes theorem is useful because it enables the calculation of the posterior probability $P(H|X)$ from the initial probabilities $P(H)$, $P(X)$, and $P(X|H)$.

3.4.2. K-nearest neighbourhood (KNN)

It is possible to use KNN, one of the simplest supervised machine learning algorithms, to estimate the class of a particular data sample by considering "feature similarity." To identify a sample, it calculates its distance from the other samples in the neighbourhood. The model's performance can be affected by the parameter k in the KNN algorithm. 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 [33, 34, 35].

The KNN classifier uses a distance function to measure the difference or similarity between two instances. Standard euclidean distance between two instances is defined in as follows:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (2)$$

x_i is the i^{th} featured factor of an example data x, while y_i is the i^{th} featured factor of an instance y, while n is representing of total features in data set.

Suppose the KNN classifiers apply set using U. In the design set, there are S samples. There are a total of L different class labels in S, so let's say $C=\{C_1, C_2,...C_l\}$ class label must be predicted for x, an input vector. This is the i^{th} vector in the design set S, and y_i denotes this. It is possible to find k vectors in S that are the closest to the input vector x using the KNN algorithm. Class C_j is assigned to the input vector x if the majority of the k closest vectors also have class C_j.

3.4.3. Decision tree

Decision tree is a non-incremental, inductive, classification algorithm. It creates a Decision tree for classifying future samples by performing a top-down, greedy search through a predefined set of examples. Each example is a member of a class and has a number of unique characteristics. The Decision tree's non-leaf nodes are all decision nodes, while the tree's leaf nodes are all class names. IDS adds a feature selection heuristic to the concept learning system algorithm. The training set of input examples is used to identify the attribute that best separates them using feature selection. It is a done deal if the attribute selected completely categorizes the training set. If IDS is applied in a greedy manner, the next best attribute is identified recursively [36, 37].

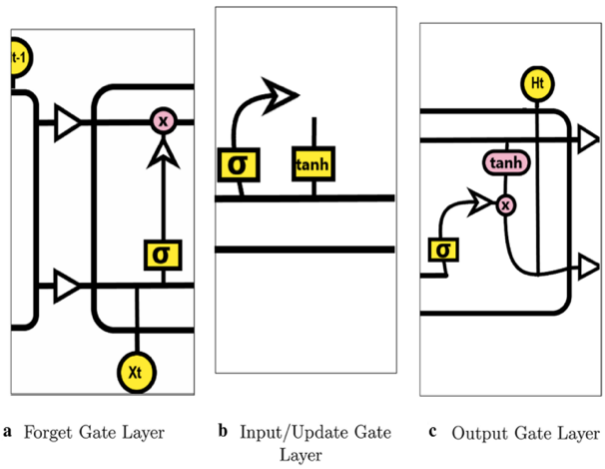
IDS uses a metric known as information gain to determine the best attribute. In information theory, entropy is used to measure the amount of information that can be gleaned from a given attribute. This attribute selection method is extremely effective. Both in the business world and in academia, IDS has a solid reputation. IDS, on the other hand, works only with examples that have the same characteristics. There must be a limited set of values for attributes. Noise or missing attributes will not be tolerated by IDS. Classes need to be clearly defined.

3.4.4. Support vector machine (SVM)

In many real-world applications, such as pattern recognition, text and image classification, hand-writing recognition, and bioinformatics analysis, Support Vector Machines (SVMs) are a powerful supervised learning algorithm that has already been successfully used in supervised learning. The concept of decision boundaries serves as the foundation for SVM classification. A decision boundary is a line that divides a collection of instances with different class values into two groups. It is capable of supporting both binary and multi-class categorizations. Each instance of the training data must belong to one of two classes in order to produce a result for IDS detection from the NSL-KDD dataset. Based on a non-probabilistic binary linear classifier, SVM classification creates a learning model that automatically assigns new instances to one of several

Table 2. List of notation description

σ	represent of standard deviation
med	represent of median operator
z_{fj}	formula of sample feature for *fj*
$max(z_f)$	maximum value representation of feature z_f
$min(z_f)$	minimum value representation of feature z_f
$\tilde{z}_{fj}$	min-max normalized value range between 0-1
z_{ij}	numeric feature value with range 0-1
z_{fj} was categorised an outlier if $z_{fj} > p \times MADE$	



a Forget Gate Layer **b** Input/Update Gate Layer **c** Output Gate Layer

Figure. 4 Basic mechanism of LSTM network [30]

classes. Due to the fact that the instances of the different classes are divided by a clear gap, SVM represents the training instances in space (wide as possible).

Once the test instances have been placed in that specific area, they are classified into one of several classes based on which side of the gap the test instance is on. SVMs are now being used for both linear and non-linear classification tasks. For mapping high-dimensional input features, it performs an efficient non-linear classification using the kernel trick, which is implemented in the code. SVM created an illustration of a linearly separable two-class classification with the two possible linear classifiers in a linearly separable two-class classification [38].

3.4.5. Deep learning using LSTM

It was developed by Hochreiter and Schmidhuber [39] in 1997 as an extension of the RNN and is intended to avoid the long-term dependency problem. Unlike RNN, LSTM is capable of retaining data for extended periods of time. The RNN architecture has a straightforward structure (e.g., a single Tanh layer), whereas the LSTM architecture is more complex, consisting of four hidden layers.

The cell state is the most important component of the LSTM. The gates are used to protect information from being added or removed from the cell state, and the sigmoid function is used to do so (one means allows the modification, while a value of zero means denies the modification.). There are three different gates that we can identify.

- Forget about the gate layer (Fig. 4 (a)): looks at the input data and the data received from the previously hidden layer, then uses a sigmoid function to determine which information LSTM will delete from the cell state (One means keeps it, 0 means delete it). It can be computed as:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (3)$$

- This layer (Fig. 4 (b)) determines which information LSTM will store in the cell state and which information LSTM will not store. The input gate layer determines which information will be updated using a sigmoid function, and then a Tanh layer proposes a new vector to be added to the cell state at the end of the cell cycle. The LSTM then updates the cell state by forgetting the information that we have decided to forget and updating it with the new vector values that we have determined to be important. It can be computed as:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \text{ and}$$

$$\tilde{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_C) \quad (4)$$

- This layer determines what will be our output by executing a sigmoid function that determines which part of the cell LSTM is going to be output, and then passing the result through a Tanh layer (value between -1 and 1) to output only the information we have decided to pass to the next neuron. The output layer is shown in Fig. 4 (c). It is calculated as:

$$h_t = o_t * tanh(C_t) \quad (5)$$

3.4.6. Hybrid SDAE with Naive bayes, Decision tree, KNN, SVM and LSTM

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 Fig. 4 below. Our experiment consists of several evaluation processes, including multi-class and binary-class using confusion matrix, accuracy, recall,

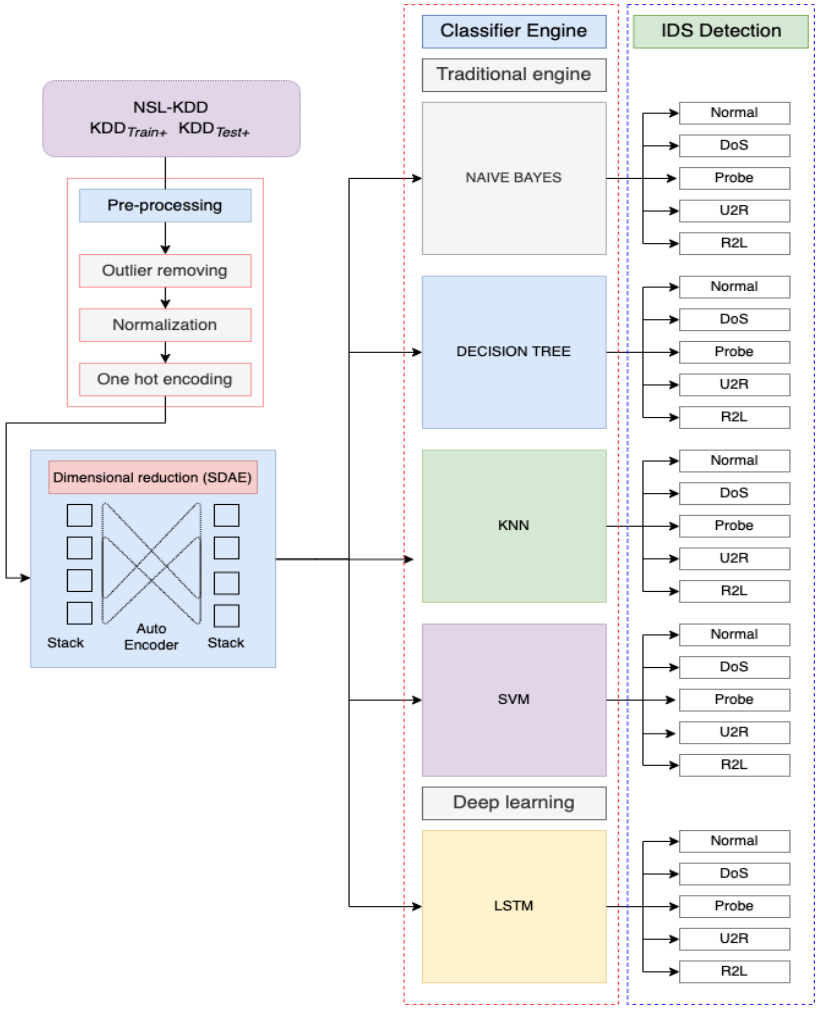


Figure. 5 Proposed model of SDAE & traditional machine

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.

We compared 4 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%. This schematic training ratio has been conducted by the majority of researchers in IDS detection.

3.5 Evaluation of IDS attack network detection

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, which indicates the number of normal samples that tested negative (accurate detection); FP represents the false positive rate, which represents how many abnormal samples tested positive (inaccurate detection); and 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 defined as 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.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{6}$$

$$Precision = \frac{(TP)}{(TP + FP)} \tag{7}$$

$$Recall = \frac{(TP)}{(TP + FN)} \quad (8)$$

$$F1 - score = 2 \times \frac{(Preciision \times Recall)}{(Precision + Recall)} \quad (9)$$

4 Result and analysis

The result of dimensional reduction using SDAE can be seen in Fig. 6 below. The dark colours represent values that are almost similar to the actual values, while the bright ones represent values that are very different from the actual values. Then, the output from dimensional reduction from SDAE would be integrated into 5 machine learning algorithm for classification task.

The evaluation metrics include accuracy, precision, recall, and F1 as shown in Table 2. The experiment of five model consisted of 2 classes which were multi-class and binary class, in which binary class only detected an anomaly and normal category, while multi-class involved 5 categories condition including "Normal", "DoS", "Probe", "R2L", and "U2R".

As shown in Table 2, 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% when compared with KNN without SDAE that 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 machine 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 that 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 that only

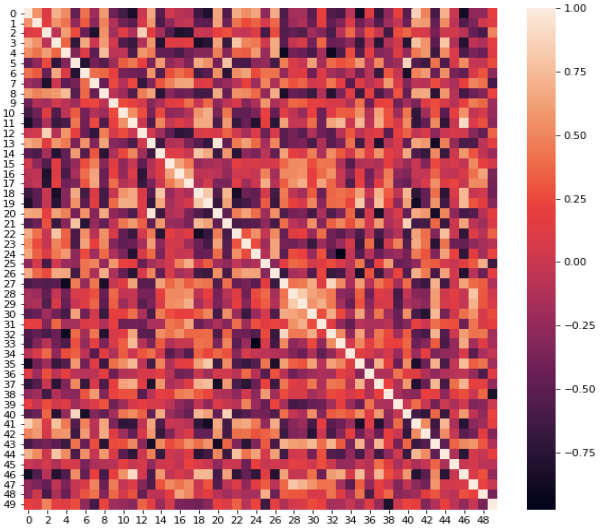


Figure. 6 SDAE training result of NSL-KDD

reached 80.1%. The best achievement in our experiment was reached by the hybridization 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.

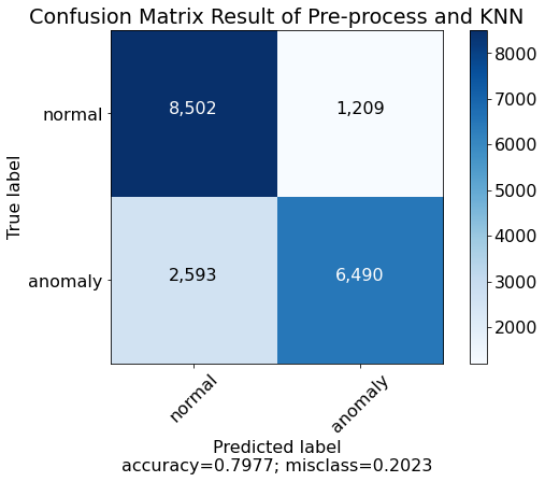
Our study also applied a confusion matrix to detect the effectiveness of our model. The confusion matrix was tried in each hybridization model and evaluated based on the multi-class and binary class classification approach. The binary class is shown in Fig. 7, while the multi-class classification can be seen in Fig. 8, where they demonstrated the involvement of SDAE, showing success in reducing error class detection in every hybridization scenario including SDAE with KNN, Naive bayes, Decision tree, SVM, and deep LSTM.

According to experiment report on Table 2, hybridization model between SDAE and KNN could increase accuracy detection by 79.8% from 77.9%. The combination between SDAE and Naive bayes achieved 80.5% while traditional pre-processing and Naive bayes only reached 76.3%. 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 83.4% while the traditional Decision tree and pre-processing only reached 82.9%. Meanwhile, the hybridization between SDAE and SVM has become the best performance with an accuracy of 84.1% in the term of traditional machine learning algorithm. The traditional pre-processing and SVM reached 80.7%.

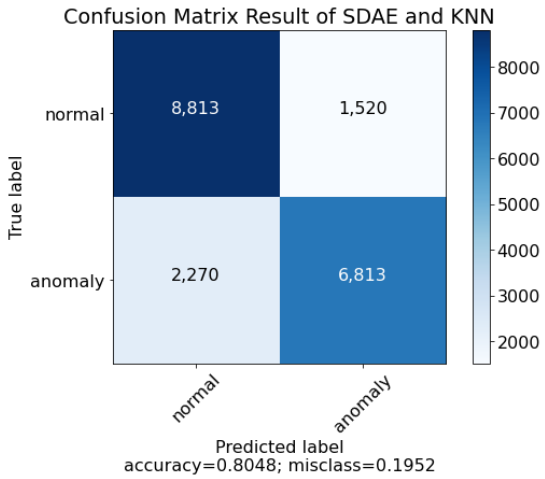
In the term of deep learning experiment point of view using LSTM as a classification engine in IDS detection also success to reach the tremendous result

Table 3. Experiment result of binary class and multi class

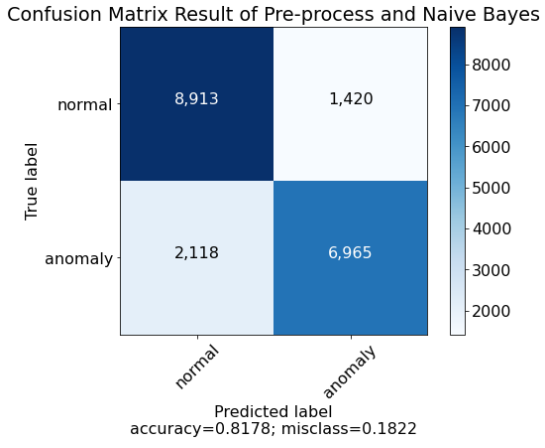
Binary classification	Accur.	Precis.	Recall	F1	Multi-classification	Accur.	Precis.	Recall	F1
SDAE & LSTM	85.2%	84.2%	82.7%	83.4%	SDAE & LSTM	84.7%	86.3%	80.1%	82.4 %
SDAE & SVM	84.1%	85.6%	83.1%	84.3%	SDAE & SVM	83.3%	85.1%	81.6%	83.3 %
SDAE & DS Tree	83.4%	83.4%	79.6%	81.4%	SDAE & DS Tree	82.8%	84.3%	80.8%	82.5 %
SDAE & NB	80.5%	81.8%	78.9%	80.3%	SDAE & NB	78.7%	80.1%	76.1%	78.0 %
SDAE & KNN	79.8%	81.1%	74.1%	77.4%	SDAE & KNN	78.1%	79.7%	75.9%	77.7 %
Pre-processing & LSTM	83.2%	80.1%	77.6%	81.3%	Pre-processing & LSTM	81.5%	83.2%	79.2%	80.2 %
Pre-processing & SVM	80.7%	81.9%	78.7%	80.2%	Pre-processing & SVM	80.0%	82.1%	77.8%	79.8 %
Pre-processing & DS3	82.9%	83.3%	81.2%	82.2%	Pre-processing & DS3	80.1%	82.7%	76.9%	79.6 %
Pre-processing & NB	76.3%	77.6%	73.8%	75.6%	Pre-processing & NB	77.8%	79.1%	75.1%	77.0 %
Pre-processing & KNN	77.9%	78.3%	75.8%	77.0%	Pre-processing & KNN	75.6%	77.1%	72.3%	74.6 %



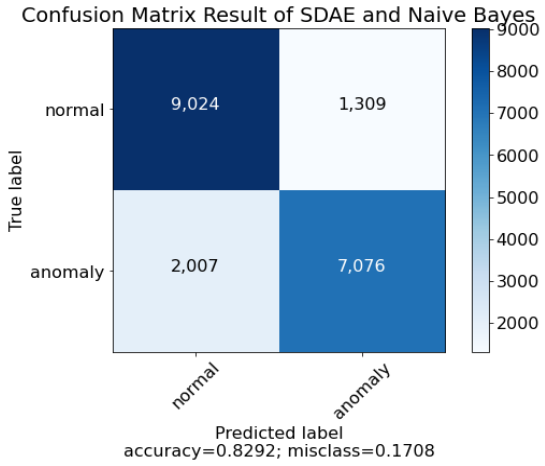
(a)



(b)



(c)



(d)

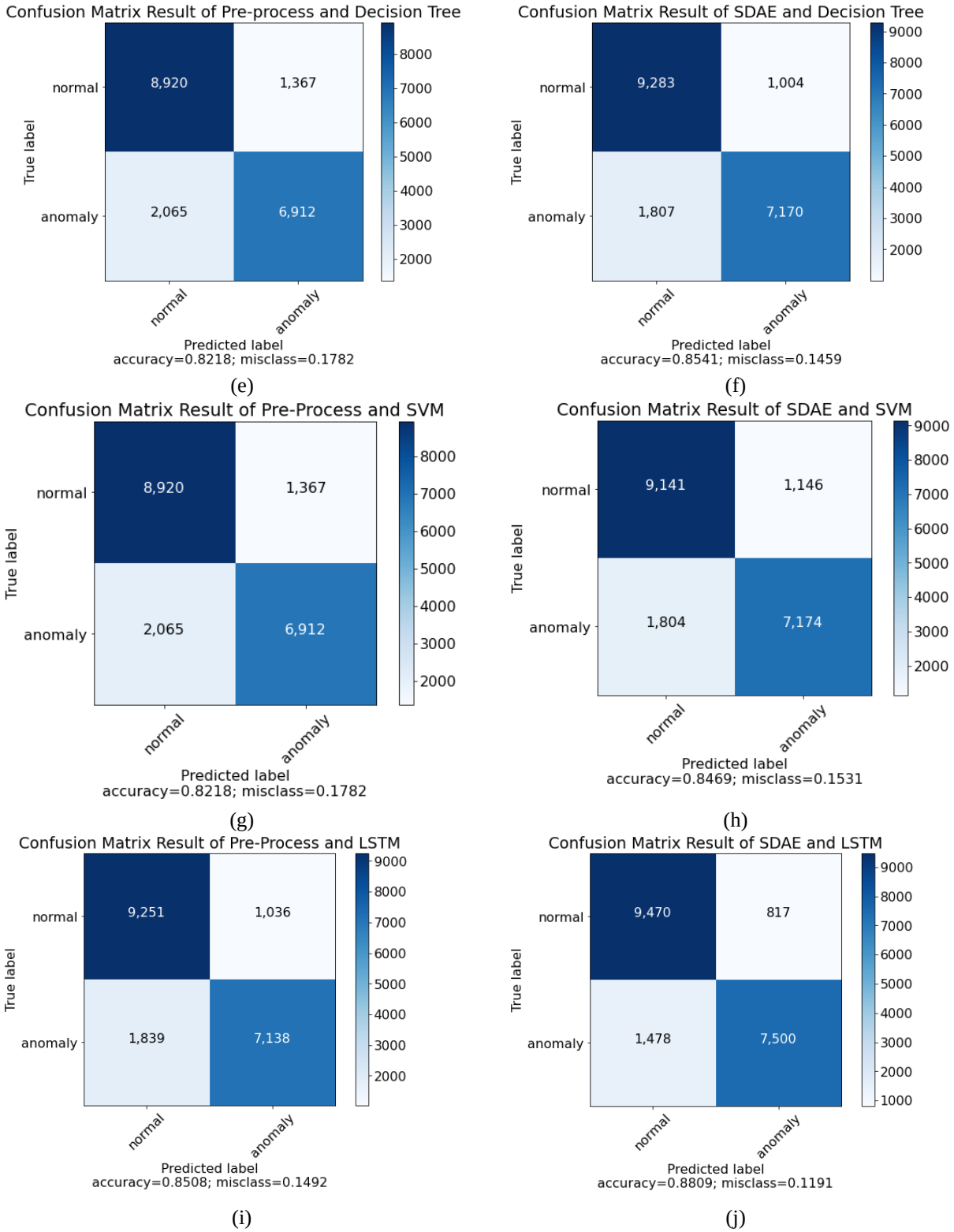


Figure. 7 Experiment scenario for binary classification

in accuracy performance by 83.2%. The enhancement dimensional reduction using SDAE reach more than 85%. The effectiveness of SDAE to enhance LSTM achieved more than 3%.

The employment of SDAE proved more effective in every hybridization scenario, according to confusion matrix in binary-class classification experiment. The combination between SDAE and KNN success to reduce error detection from 20% to 19% and increase correct detection from 79.7% to 80.4%. The effectiveness of SDAE and KNN achieved almost 2% (Fig. 7 (a) and (b)).

Fig 7. (c) and (d) demonstrated the performance

Table 4. Comparison result over state-of-the-art

No	Model	Accuracy
1	SDAE & LSTM (our model)	86.8%
2	SDAE & SVM (our model)	84.1%
3	SDAE & Decision tree (our model)	83.4%
4	SDAE & Naive Bayes (our model)	80.5%
5	SDAE & KNN (our model)	79.8%
6	CNN & LSTM (BAT) [40]	84.25%
7	Statistic & ML [41]	83.65%
8	LSTM & PCA [30]	83.8%
9	LSTM & MI [30]	83.4%

of SDAE and Naive bayes following to confusion matrix evaluation, where this model success reduces error detection from 18.2% to 17% and increase correct detection from 81.7% to 82.9%. The SDAE as a dimensional reduction application play important role in increasing performance in SDAE and Naive bayes. The effectiveness of hybridization model between SDAE and Decision tree can be seen on Fig. 7 (e) and (f). In this section, SDAE also play essential role in increasing effectiveness in error detection from 17.8% to 14.5% in increase correct detection from 82.1% to 85.4%. The hybridization between SDAE and SVM reach best performance following to traditional machine learning algorithm point of view, where this model reach for error detection in 17.8% to 15.3% in error class detection, and increase effectiveness from 82.1% to 84.6% in correct detection.

In this study, our novel model involves SDAE and LSTM. In several literatures, SDAE categorical deep learning model. It means, the propose model in this study can be inferred the hybridization between deep learning (SDAE and LSTM). The effectiveness of SDAE and LSTM can be seen on Fig. 7 (i) and (j). Indeed, this dual deep learning model achieved tremendous performance over previous experiment in this study. The LSTM classification result achieved in 85.1% and, increase performance in IDS attack detection to 88.09% when combining with SDAE in enhance dimensional reduction for NSL-KDD datasets.

The experiment report based on the confusion matrix for multi-class classification is shown in Fig. 8. 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 77.5%, while the traditional KNN and pre-processing only reached 75.8%. The combination between SDAE and Naive bayes also successful to increase performance in multi-class IDS detection in which this model achieved an accuracy of 79.9% compared to Naive bayes and pre-processing that reached an accuracy of 77.9%. The Decision tree that applied SDAE also successful to reduce miss classification and increase accuracy in confusion matrix evaluation that achieved 82.2% whereas the Decision tree without SDAE only reached 80.7%. Another hybridization model involving SDAE and SVM, evaluated using a confusion matrix, reached the best performance over the previous hybridization approach. The combination between SDAE and SVM could reduce miss classification and increase accuracy performance by 83.1% and achieve an accuracy of 81.2% in pre-processing and SVM only.

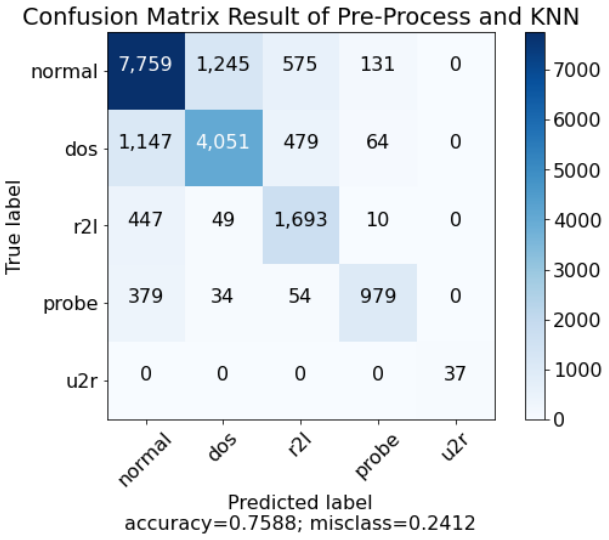
The adoption of deep learning algorithm based on LSTM as the classification engine in IDS attack detection achieved best performance in this study and become essential finding, where combining between SDAE and LSTM success to achieved high performance in 87.08%, while adoption LSTM without SDAE achieved in 84.2%. Moreover, this model also achieves effectiveness in reducing miss class detection significantly. The tremendous achievement to reduce miss class reach in 15% when applied without SDAE, while effectiveness in reducing miss class classification reach 12% when applied SDAE as a dimensional reduction engine mechanism.

The comparison result over the previous state-of-the-art has been conducted on 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 using accuracy evaluation method is shown in Table 3.

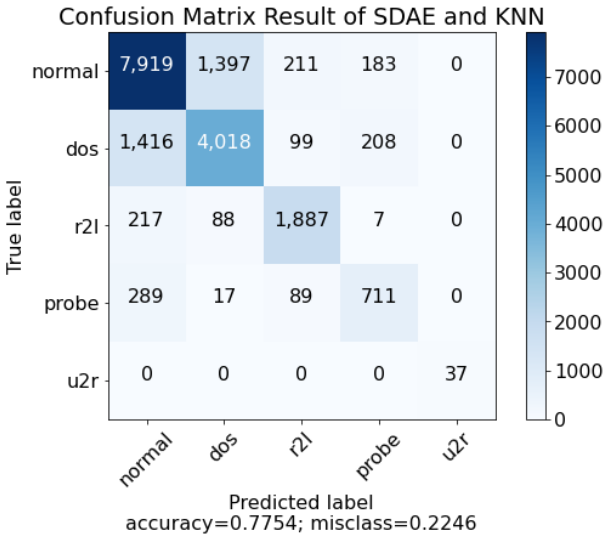
5 Conclusion and future work

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

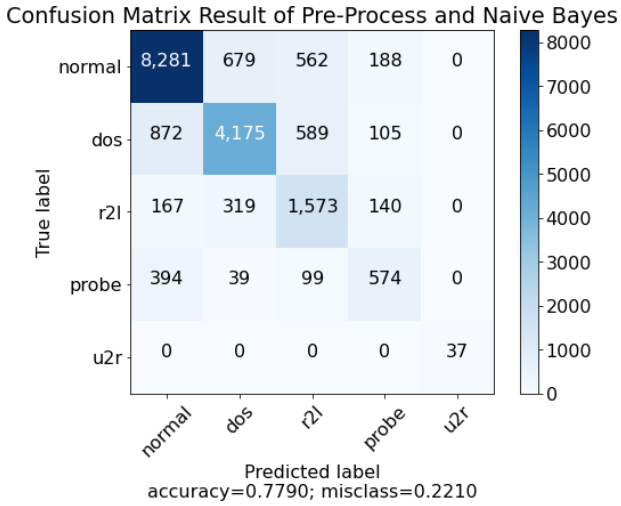
SDAE was also successful in increasing the effectiveness of classification mechanisms in machine learning especially in IDS detection even when compared to modern machine learning approaches such as deep learning based on CNN and MLP in binary and multi-class classification methods.


Confusion Matrix Result of Pre-Process and KNN
normal 7,759 1,245 575 131 0
dos 1,147 4,051 479 64 0
r2l 447 49 1,693 10 0
probe 379 34 54 979 0
u2r 0 0 0 0 37
True label
normal dos r2l probe u2r
Predicted label
accuracy=0.7588; misclass=0.2412
7000 6000 5000 4000 3000 2000 1000 0

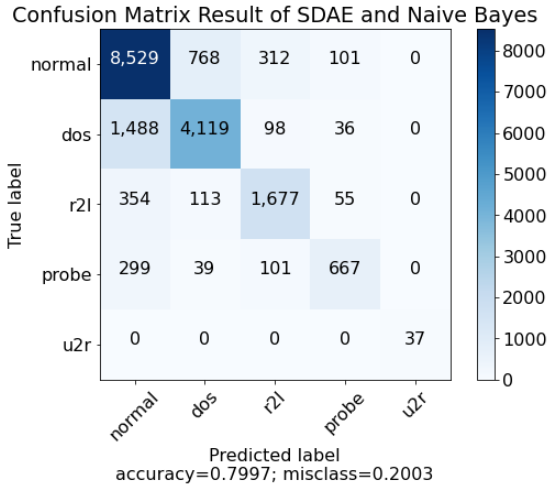
(a)


Confusion Matrix Result of SDAE and KNN
normal 7,919 1,397 211 183 0
dos 1,416 4,018 99 208 0
r2l 217 88 1,887 7 0
probe 289 17 89 711 0
u2r 0 0 0 0 37
True label
normal dos r2l probe u2r
Predicted label
accuracy=0.7754; misclass=0.2246
7000 6000 5000 4000 3000 2000 1000 0

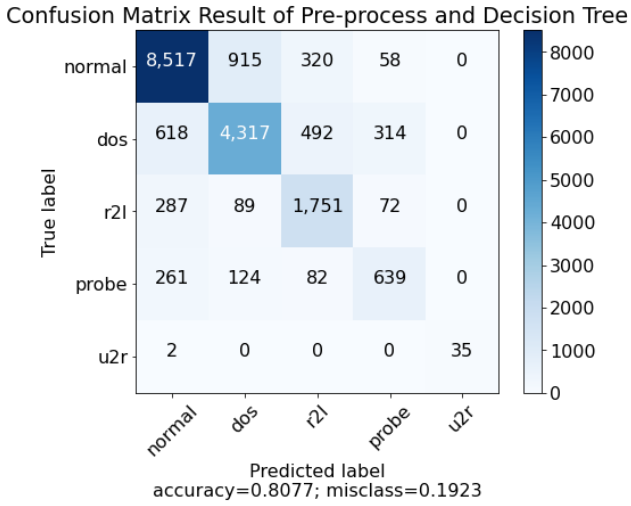
(b)


Confusion Matrix Result of Pre-Process and Naive Bayes
normal 8,281 679 562 188 0
dos 872 4,175 589 105 0
r2l 167 319 1,573 140 0
probe 394 39 99 574 0
u2r 0 0 0 0 37
True label
normal dos r2l probe u2r
Predicted label
accuracy=0.7790; misclass=0.2210
8000 7000 6000 5000 4000 3000 2000 1000 0

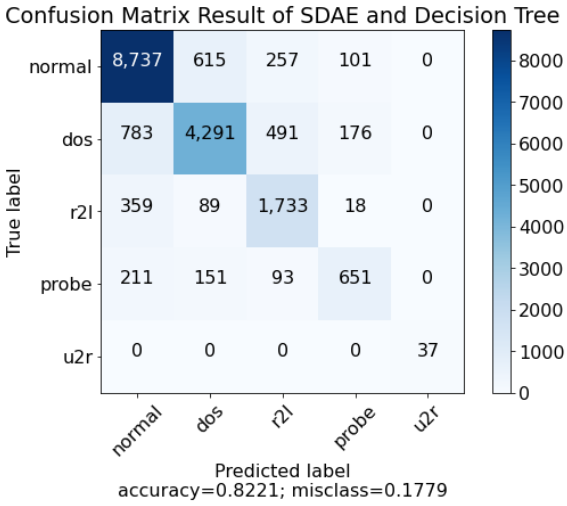
(c)


Confusion Matrix Result of SDAE and Naive Bayes
normal 8,529 768 312 101 0
dos 1,488 4,119 98 36 0
r2l 354 113 1,677 55 0
probe 299 39 101 667 0
u2r 0 0 0 0 37
True label
normal dos r2l probe u2r
Predicted label
accuracy=0.7997; misclass=0.2003
8000 7000 6000 5000 4000 3000 2000 1000 0

(d)


Confusion Matrix Result of Pre-process and Decision Tree
normal 8,517 915 320 58 0
dos 618 4,317 492 314 0
r2l 287 89 1,751 72 0
probe 261 124 82 639 0
u2r 2 0 0 0 35
True label
normal dos r2l probe u2r
Predicted label
accuracy=0.8077; misclass=0.1923
8000 7000 6000 5000 4000 3000 2000 1000 0

(e)


Confusion Matrix Result of SDAE and Decision Tree
normal 8,737 615 257 101 0
dos 783 4,291 491 176 0
r2l 359 89 1,733 18 0
probe 211 151 93 651 0
u2r 0 0 0 0 37
True label
normal dos r2l probe u2r
Predicted label
accuracy=0.8221; misclass=0.1779
8000 7000 6000 5000 4000 3000 2000 1000 0

(f)

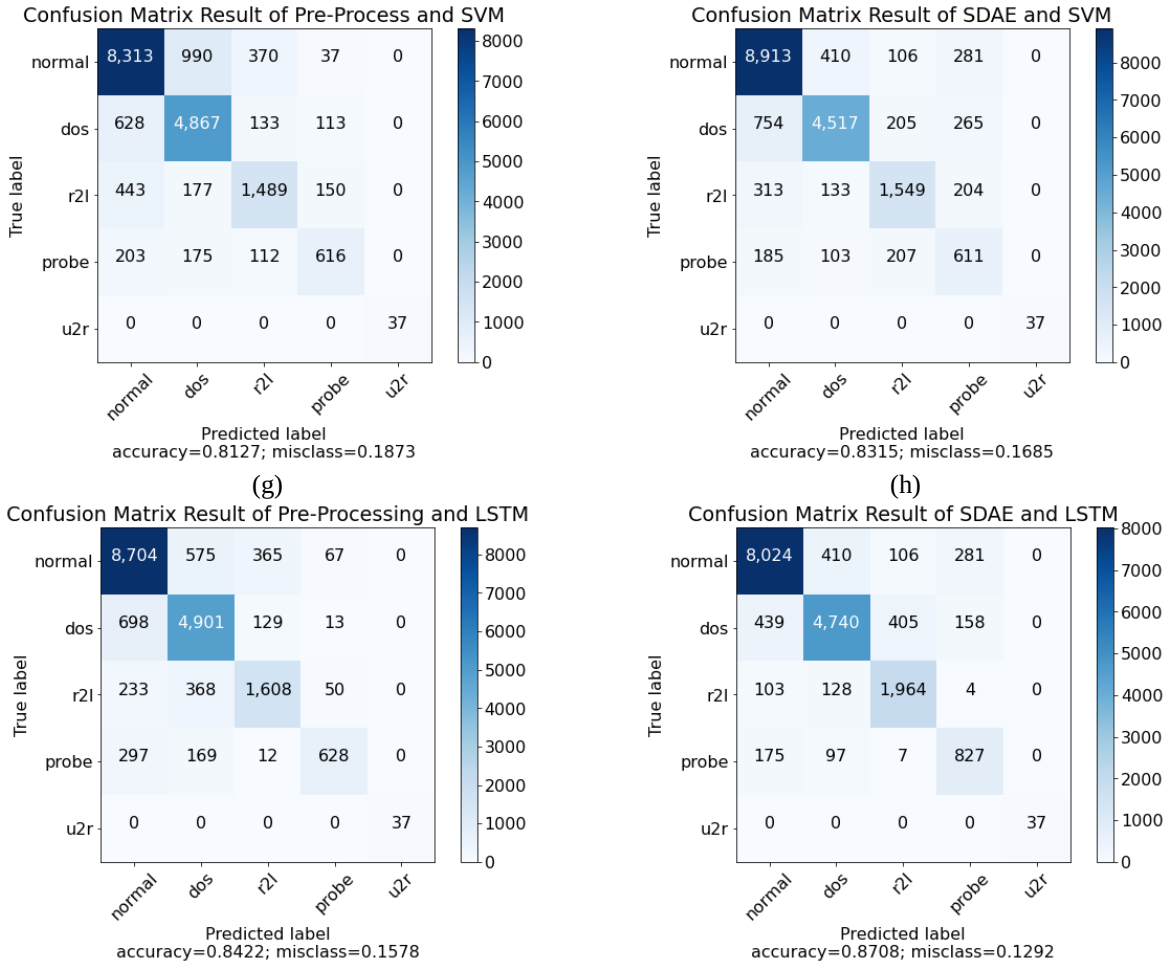


Figure. 8 Experiment scenario for multi-class classification

SDAE model aims to enhance dimensional reduction representation for IDS detection success to increase effectiveness in some traditional and modern machine learning. We believed it have impact of feature extraction mechanism and stack of dimensional data modelling by SDAE. There are some challenges in future research, in that SDAE is possible to be integrated with modern deep learning 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.

Acknowledgments

This work was supported by the Universitas Amikom Yogyakarta.

Conflicts of Interest

The authors declare no conflict of interest on this study, authorship and publishing of this manuscript.

Author Contributions

Hanafi, conveyed the idea of the system, created algorithm and formal analysis, investigation, and data preparation; Andi Sunyoto prepared and wrote the original draft, reviewed and edited the revised draft; analyzed and verified the research results and findings.

References

[1] B. Zarpelão, R. Miani, C. Kawakani, S Alvarenga, “A survey of intrusion detection in Internet of Things”, *Journal of Network and Computer Application*., Vol. 84, No 2017, pp 25-37, doi: 10.1016/j.jnca.2017.02.009.

[2] K. N. L. B. Mukherjee and L. T. Heberlein, “Network Intrusion Detection”, *IEEE Netw.*, 1994.

[3] S. Wagh, A. A. Shah, S. K. 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. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "IDS - attention : an efficient algorithm for intrusion detection systems using attention mechanism", *J. Big Data*, 2021, doi: 10.1186/s40537-021-00544-5.

[7] H. Zhang, "Design of intrusion detection system based on a new pattern matching algorithm", In: *Proc. of 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 and D. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs", In: *Proc. of the 8th International Conference on Software, Knowledge, Information Management and Application (SKIMA 2014)*, pp 1-6, 2015, doi: 10.1109/SKIMA.2014.7083539.

[11] E. Sandhya and A. Kumarappan, "Enhancing the Performance of an Intrusion Detection System Using Spider Monkey Optimization in IoT", *Int. J. Intell. Eng. Syst.*, Vol. 14, No. 6, pp. 30–39, 2021, doi: 10.22266/ijies2021.1231.04.

[12] 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.

[13] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN", In: *Proc. of Int. Conf. Signal Process. Commun. Eng. Syst. - Proc. SPACES 2015, Assoc. with IEEE*, pp. 92–96, Mar. 2015, doi: 10.1109/SPACES.2015.7058223.

[14] 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.

[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, 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.

[17] 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, pp. 1063–1070, 2019, doi: 10.18517/ijaseit.9.3.8843.

[18] Hanafi, N. Suryana, and A. Samad, "Dynamic convolutional neural network for eliminating item sparse data on recommender system", *Int. J. Adv. Intell. Informatics*, Vol. 4, No. 3, pp. 226–237, 2018.

[19] 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.

[20] 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.

[21] Hanafi, E. Pujastuti, A. Laksito, R. Hardi, R. Perwira, A. Afriandi, and Asroni, "Handling Sparse Rating Matrix for E-commerce Recommender System Using Hybrid Deep Learning Based on LSTM , SDAE and Latent Factor", *Int. J. Intell. Eng. Syst.*, Vol. 15, No. 2, pp. 379–393, 2022, doi: 10.22266/ijies2022.0430.35.

[22] 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.
[23] Q. Niyaz, W. Sun, A. Javaid, and M. Alam, "A deep learning approach for network intrusion detection system", *Eprints. Eudl. Eu*, 2016, doi: 10.4108/eai.3-12-2015.2262516.
[24] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network", In: *Proc. of International Conference on Computational Science and Engineering (CSE) and International Conference on Embedded and Ubiquitous Computing (EUC) 2017*, pp. 639-642, doi: 10.1109/CSE-EUC.2017.119.
[25] F. Qu, J. Zhang, Z. Shao, and S. Qi, "An intrusion detection model based on deep belief network", In: *Proc. of International Conference on Network, Communication and Computing (ICNCC) 2017*, pp. 97–101, Dec. 2017, doi: 10.1145/3171592.3171598.
[26] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks", In: *Proc. of National Aerospace and Electronics Conference (NAECON)*, 2015, pp. 333–344, doi: 10.1109/NAECON.2015.7443094.
[27] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network", In: *Proc. of 2017 IEEE Int. Conf. Big Data Smart Comput. BigComp 2017*, pp. 313–316, 2017, doi: 10.1109/BIGCOMP.2017.7881684.
[28] 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.
[29] K. Hara and K. Shiomoto, "Intrusion Detection System using Semi-Supervised Learning with Adversarial Auto-encoder", In: *Proc. of IEEE/IFIP Netw. Oper. Manag. Symp. 2020 Manag. Age Softwarization Artif. Intell. NOMS 2020*, 2020, doi: 10.1109/NOMS47738.2020.9110343.
[30] 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.
[31] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", In: *Proc. of IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, No. July, 2009, doi: 10.1109/CISDA.2009.5356528.
[32] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", *J. Mach. Learn. Res.*, Vol. 11, pp. 3371–3408, 2010, doi: 10.1111/1467-8535.00290.
[33] 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.
[34] 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.
[35] 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.
[36] 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.
[37] 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, [Online]. Available: https://www.researchgate.net/publication/298175900.
[38] M. A. Qatf, Y. Lasheng, M. A. Habib, and K. A. Sabahi, "Deep Learning Approach Combining Sparse Autoencoder with SVM for Network Intrusion Detection", *IEEE Access*, Vol. 6, No. c, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.
[39] S. Hochreiter and J. U. Schmidhuber, "Lstm", *Neural Comput.*, Vol. 9, No. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[40] 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.
[41] 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.