# IDSX-Attention: Intrusion detection system (IDS) based hybrid MADE-SDAE and LSTM-Attention mechanism

Hanafi Hanafi [a,1,*], Andri Pranolo [b,2], Yingchi Mao [c,3], Taqwa Hariguna [d,4], Leonel Hernandez [e,5], Nanang Fitria Kurniawan [f,6]

[a] Informatics Department, Universitas Amikom Yogyakarta, Indonesia
[b] Informatics Department, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[c] College of Computer and Information, Hohai University, China
[d] Department Information System, Universitas Amikom Purwokerto, Indonesia
[e] Faculty of Engineering, Institución Universitaria de Barranquilla, Colombia
[f] Department of Information System, Institut Teknologi Tangerang Selatan, Indonesia
[1] hanafi@amikom.ac.id; [2] andri.pranolo@tif.uad.ac.id; [3] taqwa@amikompurwokerto.ac.id; [4] taqwa@amikompurwokerto.ac.id;
[5] lhernandezc@unibarranquilla.edu.co; [6] nanangfk@itts.ac.id
* corresponding Author

ARTICLE INFO

**Article history**
Received November 7, 2022
Revised February 4, 2023
Accepted February 14, 2023
Available online March 31, 2023

**Keywords**
IDS
Cyber security
Attention mechanism
SDAE
LSTM

ABSTRACT

An Intrusion Detection System (IDS) is essential for automatically monitoring cyber-attack activity. Adopting machine learning to develop automatic cyber attack detection has become an important research topic in the last decade. Deep learning is a popular machine learning algorithm recently applied in IDS applications. The adoption of complex layer algorithms in the term of deep learning has been applied in the last five years to increase IDS detection effectiveness. Unfortunately, most deep learning models generate a large number of false negatives, leading to dominant mistake detection that can affect the performance of IDS applications. This paper aims to integrate a statistical Median Absolute Deviation Estimator (MADE) to remove outliers in pre-processing, Stacked Denoising Autoencoder (SDAE) is responsible for reducing data dimensionality, and LSTM-Attention is responsible for producing attack classification tasks. The model was implemented into NSL-KDD dataset and evaluated using Accuracy, F1, Recall, and Confusion metrics. The results showed that the proposed IDSX-Attention outperformed the baseline model, SDAE, LSTM, PCA-LSTM, and Mutual Information (MI)-LSTM, achieving more than a 2% improvement on average. This study demonstrates the potential of the proposed IDSX-Attention, particularly as a deep learning approach, in enhancing the effectiveness of IDS and addressing the challenges in cyber threat detection. It highlights the importance of integrating statistical models, deep learning, and dimensionality reduction mechanisms to improve IDS detection. Further research can explore the integration of other deep learning algorithms and datasets to validate the proposed model's effectiveness and improve the performance of IDS.



## 1. Introduction

The increasing number of internet users worldwide influences millions of devices connected to the internet. As a result, the data stored on our personal computers has significantly increased in value. Additionally, as more businesses allow employees to work from home, networks are more vulnerable to information theft and loss. Additionally, since access to the internet network is widely available and inexpensive, anyone who engages in cybercrime worldwide can launch a network attack regardless of

where they are physically located. Attacks on networks are unlawful intrusions into private networks intending to damage, destroy, or steal information.

Moreover, the scope and depth of security technologies for computers and networks continue to increase in response to the evolving nature of the threats they face. The intrusion detection system is the most crucial of these aspects [1]. Intrusion detection systems (IDSs) are essential to any comprehensive security architecture. An IDS's overarching goal is to keep screening for any signs of malicious behavior or previously identified risks in network data. IDS alerts the IT administrator to a possible network intrusion after a detected threat. The IP address of the intruder, the address of the victim system, and the suspected type of attack will often all be included in the reported data. Several problems can arise with intrusion detection systems, including a high proportion of false positives and false negatives. In other words, it was a false positive. This situation arises when an IDS incorrectly identifies a legitimate action as an attack. False positives are errors, although they typically do not significantly damage the network. False negatives, on the other side, occur when the IDS fails to spot an attack. It happens when an attack is mistakenly labeled as acceptable by the IDS. Since IT personnel are unaware of the attack, this is the most perilous stage. The context of an intrusion detection system is a simple classification problem. Many other approaches have been implemented in network security since Denning [2] initially presented the IDS. In addition, numerous deep learning strategies have been applied to intrusion detection due to the constant growth of big data and the rise in computing capacity. Since deep learning can effectively process large data sets and extract meaningful features from unprocessed data, it has become a focal point of study for many academics, involving traditional machine learning and RNN model [3][1].

Many studies have found that deep learning, traditional machine learning, and pattern similarity are the main methods used in the intrusion detection model. In recent years, deep learning's popularity has skyrocketed to the point where it has eclipsed any competing approach. Historically, intrusion detection was the primary application of pattern similarity models. Most of these models use attribute similarity to create patterns similar to their primary core learning algorithm [4][5]. Most frameworks have previously been put into use. For instance, the Knuth Morris Pratt (KMP) model, Boyer Moore (BM), Boyer Moore Harspool (BMH), Boyer Moore Harspool Sunday (BMHS), Aho-Corasiek (AC), and AC-BM model were all classic models for constructing IDSs. Experiments led to the discovery of an efficient algorithm for calculating pattern similarities in less time. Nonetheless, there is a major defect in the standard model of pattern similarity. They can't wrap their heads around the concept of intrusion detection. An important goal of this research is the creation of a cheap algorithm that can significantly cut down on both false positives and the time and money spent on them.

A hybrid model using KNN and several conventional machine learning methods found that they significantly enhanced an earlier model. K-means clustering algorithm to group similar data points and a KNN classifier [6]. CANN is the cutting-edge IDS intelligence machine for malware detection that emerged from this model. Another set of studies [7][6] proposes combining a conventional classifier with Random Forest to boost CANN performance. The hybrid model, which relied on Random Forest as its primary classifier machine, achieved superior results to the competition with an accuracy of 94.7%. It has been proposed [8] that a neural network be used to enhance a Random Forest's already impressive capabilities. When applied to NSL-KDD, the ANN model achieved an accuracy of over 81% and a classification rate of 79% for malicious detection and network attack classification, respectively. Based on NSL-KDD, a Decision Tree (DT) intrusion detection model has been proposed [9] Based on the results of the experiment, which were made public online, it was concluded that DT performed well on the IDS detection classification task. According to the preceding explanation, the detection of IDS threats is greatly improved by enhancing traditional machine learning. Many of them, however, called for elaborate attribute extraction and massive pre-processing. Unfortunately, there is no way to manage massive amounts of intrusion data using a machine learning classification approach. A deep learning model was created by fusing Deep Belief Networks (DBNs) with probabilistic neural networks [10]. DBN's work is to transform raw data from its low-dimensional representation into a non-linear one without losing any of its essential qualities. The learning in the hidden layer is improved with the help

of particle swarm optimization. Last, probabilistic neural network (PNN) detection is used in IDS detection. Experiments showed that DBN-PNN achieved an accuracy rate of 93.25%. Also, compared to earlier studies that used a PCA mechanism to boost dimensional reduction and a Probabilistic Neural Network, DBN-PNN performed better (PNN).

In contrast, another researcher [11][12] proposed a deep learning model for the IDS task using Deep Belief Networks (DBNs). This model incorporates the following two essential processes. The process can be explained as follow: 1) Their model used a Restricted Boltzmann Machine (RBM) to learn each layer separately, and 2) They used backpropagation to determine the hidden layer vector from the known layer vector. The next layer's representation is the hidden layer's vector manifest. Both procedures take the final RBM output vector as input and use backpropagation networks generated by that method. With the DBM model, they can obtain a precision of 95.25 percent. Compared to backpropagation, this improves performance by 89.07%, and compared to support vector machines (SVM), this improves performance by 91.36%. DNN stands for Deep Neural Network, a type of neural network that has shown promise for use in IDS [13]. It is the DNN algorithm's take on an auto-encoder with four hidden layers and a hundred hidden units. Rectified Linear Units (ReLU) are used to engage the covert layer. ReLU can classify activation functions with non-linear behavior. This activation function aims to enhance the algorithm's capability to recognize patterns in data and perform complex classification tasks. In order to access the stochastic optimizer, this research employed the adaptive moment mechanism. The experiment results showed that DNN could achieve a measurement accuracy of 99%.

Convolutional Neural Networks (CNNs) have been proposed as a novel model for IDS network detection [14]. As a general-purpose solution, the CNN model excels at many imaging issues. In this particular IDS detection scenario, it assumed that the data vector dimension of the image processing problem is similar to that of the IDS problem. Convolutional neural networks (CNNs) are feedforward neural network that employs convolutional processes to flatten high-dimensional data sets into more manageable vectors. Using a CNN model, the authors of this work claim that their method not only reduces the false alarm rate but also increases the class's accuracy even when the sample size is small. The experiment results from the report that CNN can perform at a 79.48% accuracy level in KDD-NSL. They compared to several other traditional machine learning methods. It performs better. With the adoption of GAN (Generative Adversarial Network) and AE methods, a novel IDS detection model was proposed [15]. They used a semi-supervised model on the NSL-KDD dataset to improve IDS malicious detection without labeled data while decreasing the time and effort needed to tag the labeled data manually. Even though only 0.1% of the datasets contained labeled data, the experiment report demonstrating the use of GANs and AEs to enhance IDS detection on NSL-KDD datasets was successful. A proposed model with an enhanced dimensional reduction to improve the performance of IDS detection [16] successfully combined PCA and Chi-square with LSTM to reduce false negative rates. The dimensional reduction process is essential in increasing performance in the PCA model. Further, to optimize the reduction, UMAP and Mutual Information (MI) were used as dimensional reduction to enhance classification tasks for IDS based on the Attention mechanism [17].

CNN is a subclass of machine learning algorithm that brings success stories in the enhancement of some applications in computer science, such as image processing cases [18], natural language processing for recommender systems [19][20][21][22], and text mining [22][23]. Several IDS research applied CNN to improve classification tasks. For example, combined CNN and LSTM could reduce huge false negative detection [24], improving the classification task's effectiveness. Evaluation of Hybrid PCA/LSTM [25] shows that the DL-IDS achieves an accuracy of 98.67%. While LSTM is in charge of classifying network attacks, PCA is responsible for compressing raw data attacks. As for multiclass classification, PCA-LSTM can get as high as 99.39% accuracy while still achieving 99.45% accuracy in binary class. The performance of the LSTM was enhanced by decreasing the number of dimensions in the PCA model. Both mutual information (MI) and long short-term memory (LSTM) were proposed as a result of their study. It has a 96.24% success rate in identifying binary classes and a 95.56% success rate in identifying multiple classes. However, from the previous research, developing the deep learning to improve performance by reducing false negative for IDS applications is challenging.

In this research, we contributed by developing a novel model of IDS detection using a statistical approach to pre-processing data stage mechanism. Then, the dimensional reduction of data using Stack Denoising Auto Encoder (SDAE) is used as a sub-class of Auto Encoder that enhance feature selection and feature extraction, and involvement transformer to enhance classification method on IDS task. Moreover, we adopted NSL-KDD datasets as the most popular IDS datasets to observe our proposed method's performance.

## 2. Method

This study involves several hybridization scenarios, including classical machine learning based on Naïve Bayes, Gaussian Naïve Bayes, LSTM, and LSTM Attention. Our proposed model, called IDSX-Attention, is a deep learning approach using sequential to the sequential mechanism that is famously called Transformer to improve pre-processing method before employing it within SDAE to advance dimensional reduction. The following sections provide an in-depth analysis of the data and research techniques used. The experiment scenario of SDAE and Attention mechanism is shown in Fig. 1.

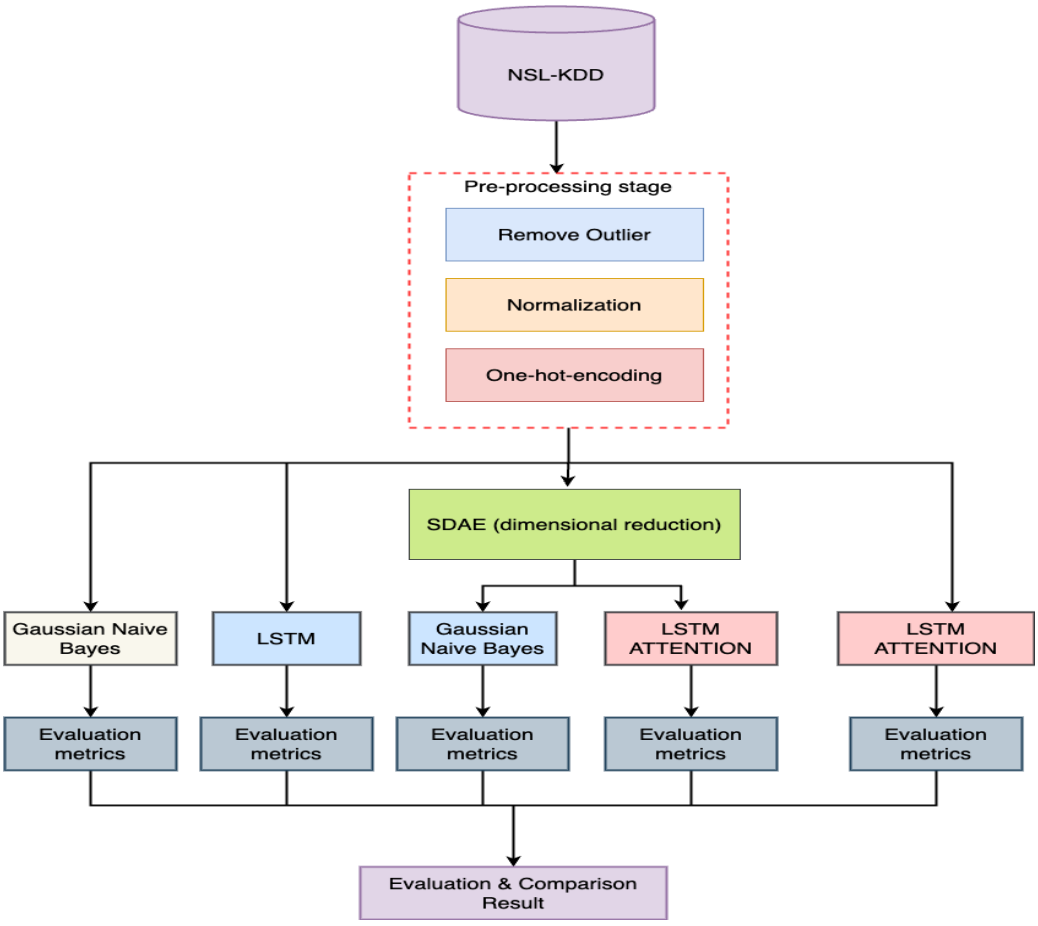


**Fig. 1.** Experiment scenario of SDAE and Attention mechanism

### 2.1. IDS Dataset using NSL-KDD

As a more advanced version of the KDD99 datasets, NSL-KDD is available for use. Several IDS network applications and studies extensively utilize this dataset to compare the efficacy of various approaches. NSL-KDD fixes several problems with the original KDD99 datasets from a replication and reiteration standpoint. Documentation of this condition is present in both test and practice sets. It will be biasing the classification engine toward the more common sample size.

NSL-KDD research is conducted to make its findings available to the public at large. Canada Cyber Security (CSS) developed NSL-KDD datasets [26]. $KDD_{Train+}$ and $KDD_{Test+}$ are two of the most

important subsets of this data set. This data set includes 125973 training samples and 22544 test samples. With the release of $KDD_{Test+}$, 17 attack types were introduced that are not part of $KDD_{Train+}$. Many researchers and academics have eliminated 3751 unnecessary attack-type categories to standardize the classification output of IDS detection. In conclusion, there are 18793 items in the KDDTest+ after subtracting 22544 from 3751. The $KDD_{Train+}$ and $KDD_{Test+}$ specs are laid out in Table 1. There are 3 symbol categories and 38 continuous attribute categories in NSL-KDD with feature *zf(f=1,2,3,4,5,..41)* feature. The NSL-KDD datasets divided into 4 class attack as follow:

- **Deniel of Service (DoS)**: If someone is attempting to prevent users from accessing a particular network service, server, or other services on the internet, they are launching a DoS attack. DoS attack, unauthorized users disrupt or slow down a network's servers or services.
- **Remote to User (R2L)**: During an R2L attack, a hacker will send spoofed remote packets to a server or computer system to gain unauthorized access.
- **User to Root (U2R)**: U2R refers to a class of attacks that targets a computer's "root" directory. To exploit this vulnerability, the hacker first gains access to the system under the guise of a legitimate user.
- **Probe**: Uncontrolled data collection from a network or security management system is called a probe attack.

The full breakdown of NSL-KDD characteristics, including the total number of records for each type of attack, can be found in Table. 1 and sub-class attack profile was detailed and described in Table. 2.

**Table 1.** NSL-KDD dataset characteristic

| 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 |

**Table 2.** Sub-attack class category

| No | DoS | Probe | U2R | R2L |
|---|---|---|---|---|
| 1 | worm | satan | xterm | httptunnel |
| 2 | teardrop | saint | sqlattack | Guess_passw |
| 3 | smurf | portsweep | rootkit | ftp_write |
| 4 | udpstorm | mscan | ps | multihop |
| 5 | pod | nmap | perl | imap |
| 6 | processtable | ipsweep | loadmodule | phf |
| 7 | mailbomb | | Buffer_overflow | named |
| 8 | neptune | | | spy |
| 9 | land | | | snmpgetattack |
| 10 | back | | | sendmail |
| 11 | Apache2 | | | Snmpguess |
| 12 | | | | warezclient |
| 13 | | | | warezmaster |
| 14 | | | | xlock |
| 15 | | | | xsnoop |
| Total / class | 11 | 6 | 7 | 15 |
| Total class | | 39 | | |

### 2.2. Statistical and Data Pre-processing

We adopt a pre-processing method using a statistical approach [27]. The pre-processing of NSL-KDD datasets aims to standardize the transformation of the raw data. It can therefore be effective and suitable for the subsequent section's process. It also attempts to make sure that the machine learning process can distinguish the property of the assault characteristic. The pre-processing stage is divided into three sessions. Following is a detailed explanation of each statistical pre-processing step.

#### 1) Outlier removal

We must eliminate a value from the NSL-KDD because it is inconsistent. These people commonly refer to this issue as the "outlier problem". A necessary step must precede the normalization of the data. Outliers may impact the suggested methodology for detecting malicious behavior, which could result in inaccurate detection. The following computation is the basis for the Median Absolute Deviation Estimator (MADE) approach.

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

where *med* represents the median calculator, while zfj is the representative of feature zf , and variable P=1.4826 indicates a constant value below the estimation of normal data. The value of $z_{fj}$ estimated as outlier if:

$$zfj > p^{*}MADE \quad (2)$$

#### 2) Data normalization

In this research, the min-max approach to the normalization process aims to transform the numeric value $z_{fj}$ into a numeric range value between 0-1. The formula is as follows:

$$\tilde{z}_{fj} = \frac{z_{fj} - \min(z_f)}{\max(z_f) - min(z_f)} \quad (3)$$

where variable $\max(z_f)$ is represent the maximum value and $\min(z_f)$ is represent the minimum value of $\int^{th}$ feature $z_f$ while $\tilde{z}_{fj}$ is the result of a normalization value in the range 0-1.

#### 3) One-hot-encoding process

There are three types of class attack include protocol number, service number, and flag number ($z_2$, $z_3$, $z_4$) that convert into numeric numbers based on the one-hot-encoding method. Especially for every categorical feature that is denoted by a binary number. From the network application perspective, protocol services, such as *tcp*, *udp,* and *icmp* apply to one-hot-encoding in the term of a binary vector.

**Table 3.** Result of NSL-KDD after removing outlier

| Dataset pre-process | Profile of attack | | | | | |
|---|---|---|---|---|---|---|
| NSL-KDD | Total | Normal | DoS | Probe | R2L | U2R |
| $KDD_{Train+}$ | 125973 | 67343 | 45927 | 11656 | 995 | 52 |
| *After remove outlier* | 85421 | 51551 | 23272 | 9683 | 974 | 41 |
| $KDD_{Test+}$ | 18793 | 9710 | 5741 | 1106 | 2199 | 37 |
| *After remove outlier* | 11925 | 7341 | 1975 | 620 | 1971 | 18 |

### 2.3. SDAE as dimensional reduction

An autoencoder, often known as an AE, is a specialized kind of neural network that accepts information as its input, then uses deterministic mapping to map or encode that information to a hidden representation. In order to learn a mapping technique from data, denoising autoencoders recreate the input from a distorted version of the data. Additionally, denoising autoencoders can be stacked to create

a deep network, often called a stacked denoising autoencoder, which enables learning higher-level representations [28]. The illustration of the SDAE model to enhance the dimensional reduction of NSL-KDD datasets can be seen in Fig. 2.

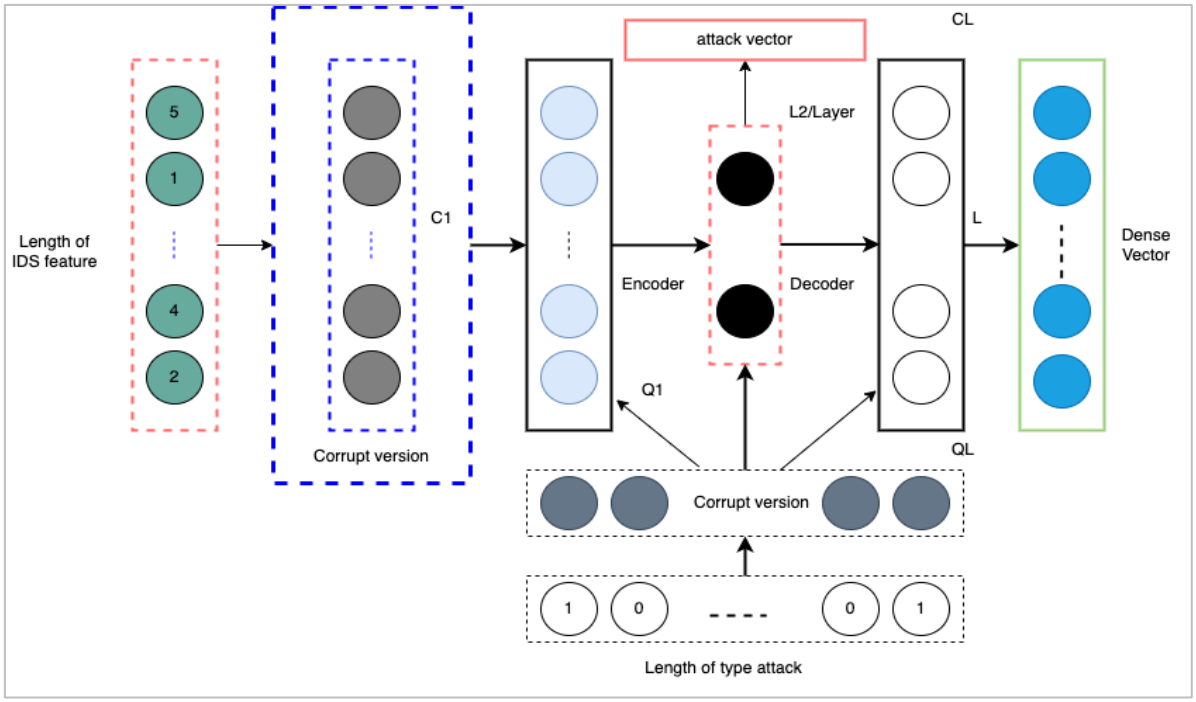


**Fig. 2.** SDAE as dimensional reduction task for NSL-KDD

where $l \in 1, \ldots., \ldots., \ldots. L-1, h_l$ as hidden layer representation can be computed with:

$$h_l = g(C_l h_{l-1} + Q_l \tilde{X} + b_l) \tag{4}$$

$C_l$ and $Q_l$ are represent the weight parameter of every layer, while $b_l$ as represent vias vector for every layer and $g()$ as non linear activation function, $h_0$ is a representation of a corrupt version of $R_i$ and $\tilde{X}$ is represent a corrupt version of $X$. While output layer $L$ can be calculated with formulas as follow:

$$\hat{R}_i = f(C_L h_L + b_{\hat{R}_i}) \tag{5}$$

$$\hat{X} = f(Q_L h_L + b_{\hat{X}}) \tag{6}$$

where the non-linear activation function is represented by $f()$. The result of the dimensional reduction process using SDAE on NSL-KDD can be seen in Fig. 3.

### 2.4. LSTM

Recurrent neural networks, or RNNs, are a type of artificial neural network in which output from one layer is used as input for the next. However, gradient disappearing and exploding difficulties during backpropagation are RNNs' fundamental weaknesses. Long Short-Term Memory (LSTM) was created in 1997 by Hochreiter and Schmid Huber [29] to address this issue. Long Short-Term Memory (LSTM) networks are a variant of traditional recurrent neural networks that can remember and use data from previous time steps. When transferring data through an LSTM network, cell states are employed instead of the more common feedforward neural networks. LSTMs can recall and forget information selectively. In this way, the RNN's gating mechanism helped solve the short-term memory issue. Units of a long short-term memory (LSTM) network are built from cells, gates (input and output), and a final gate (forget) that is connected to the network. Information is stored in the cell for an undetermined amount of time, and the three gates control the constant inflow and outflow of data.

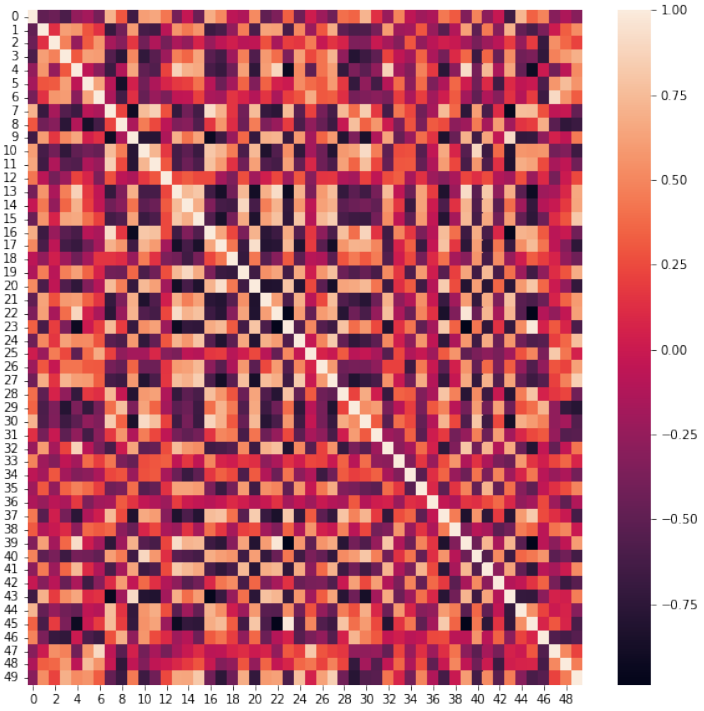

**Fig. 3.** Result of SDAE dimensional reduction

Long short-term memory (LSTM) networks are a subset of neural networks and a feedforward neural network operating on categories. Consideration of a connection between the past information section and the current section process is an advantage of the LSTM method. According to Natural Language Processing (NLP), this is a crucial step in preserving the meaning of words and phrases in their original settings. The input layer, output layer, hidden state, and preceding process are all components of the LSTM's hidden phases. Because LSTM performance involves some crucial calculations in a hidden stage, one of its benefits is the ability to launch sequential aspects. For a closer look at the LSTM model's algorithm can be seen in Fig. 5.

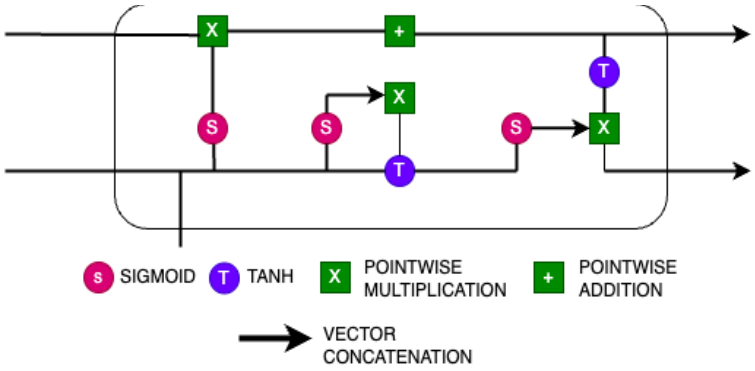


**Fig. 4.** The architecture of LSTM

### 2.5. Attention mechanism

In order to aid in the memorization of lengthy source sentences for neural machine translation, Bahdanau [30] proposed the attention mechanism. The attention mechanism builds shortcuts between the context vector and the full input stream to speed up processing rather than building a single context vector. The relative importance of these short circuits can be changed for each distinct output characteristic (see Fig. 3). The effect boosts the relevant aspects of the incoming data while diminishing the less relevant ones. Basic attention mechanism is shown in Fig. 5.

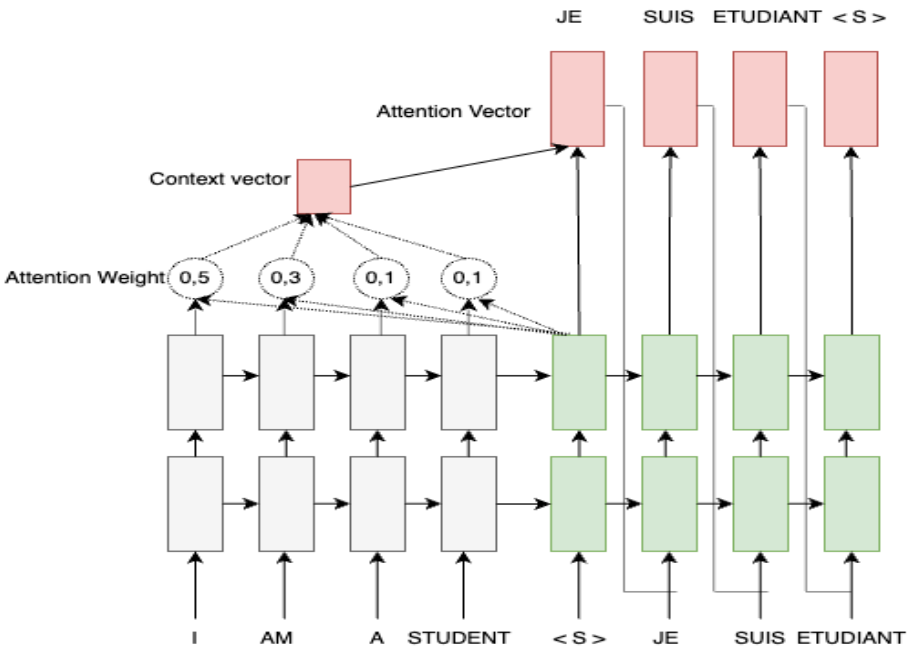


**Fig. 5.** Basic Attention mechanism architecture

With its encoder-decoder structure, the Attention model is comparable to other neural sequence conduction models. The input symbolic representation sequence ($x^1$, $x^2$,..., $x^n$) is transformed into a continuous representation sequence $z$ = ($z^1$, $z^2$,..., $z^n$) by the encoder, and the output continuous representation sequence $z$ = ($z^1$, $z^2$,..., $z^n$) is then transformed back into the input symbolic representation sequence by the decoder ($y^1$, $y^2$, . . . , $y^m$). The model is fully autoregressive, except for the input to the first encoder in the encoder stack. This implies that the output of each encoder and decoder is used as the input to the next stage in the model. An Attention model aims to convert the feature sequence fed into a vector representation that can be used in the model. On the other hand, attention's self-attention mechanism does away with the recurrent link in favor of attention matrices.

Given that not all inputs can be used to produce the same output, we calculate multiple attention weights for each one. The attention mechanism must therefore determine the number of attention weights. A context vector $C_i$ is constructed for the final outcome $y_i$ by summing the annotations and assigning weights to each. The attention mechanism determines several attention weights denoted by $\alpha(t,1), \alpha(t,2), \alpha(t,3), \ldots . \alpha(t,t)$ because not all inputs are utilized to produce the appropriate output. The weighted total of the annotations is used to generate the context vector $C_i$ for the final result $y_i$:

$$C_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{7}$$

The normalized output score of a feedforward neural network, where a function captures the alignment between input at j and output at i, and $e_{ij}$ is used to calculate the attention weights. A softmax function, defined as follows, is used to determine the weights $\alpha_{ij}$:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \tag{8}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{9}$$

where $e_{ij}$ is the result of the feedforward value from the neural network obtained by the calculation. That means to catch the correspondence between input $j$ and output from variable $i$.

### 2.6. Evaluation method

Evaluation metrics such as Precision, Recall, F1 score, and accuracy are performed to observe the proposed model's performance. The basic concept of formula evaluation is as follows:

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

$$F1\ score = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{12}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{13}$$

where TP (True Positive) epresents the total number of anomalous occurrences recognized, while TN (True Negative) indicates the rate of normal occurrences that were incorrectly classified as negative. FP (False Positives) denotes how often typical traffic patterns are incorrectly labeled as abnormal, and FN (False Negative) represents the rate of mistakenly labeling outlier patterns as normal.

## 3. Results and Discussion

Some train scenarios implement without dimensional reduction, and another section training process learns by dimensional reduction based on SDAE. First, traditional machine learning using Gaussian Naïve Bayes (GNB) without SDAE achieved lower performance over GNB with SDAE in 80.8%. While LSTM, as a modern sequential machine learning train without SDAE, achieved lower performance over LSTM, and included dimensional reduction based on PCA (Principle Component Analysis), MI (Mutual Information), and SDAE. Applying dimensional reduction success can be inferred to enhance IDS application effectiveness where MI and LSTM achieved 82.2%, PCA and LSTM achieved 83%, and SDAE and LSTM achieved 83.9%. Our second model uses SDAE, LSTM, and applying Attention mechanism success to increase the effectiveness of IDS classification task in 84.8%. That means enhancement of SDAE and LSTM using Attention mechanism success to increase the effectiveness of IDS application by more than 2%. Moreover, SDAE, LSTM, and Attention mechanism called IDSX-Attention achieve better Accuracy, Precision, Recall, and F1 measure performance than the other models (Table 4).

**Table 4.** Experiment and comparison result

| Hybrid model | Accuracy | Precision | Recall | *F1* |
|---|---|---|---|---|
| **SDAE+LSTM+ATT** | **0.849906411071** | **0.623722910881** | **0.669476568698** | **0.645790352979** |
| SDAE+LSTM | 0.839739859104 | 0.624520242214 | 0.6770094037055 | 0.649706409852 |
| PCA+LSTM | 0.830591154169 | 0.613412139719 | 0.6701998971287 | 0.628976731932 |
| MI+LSTM | 0.822091831791 | 0.621034194378 | 0.6691875194568 | 0.639897184189 |
| LSTM Only | 0.821689476966 | 0.652813196182 | 0.6803603172302 | 0.666302156138 |
| SDAE+Gausian NB | 0.808159154723 | 0.705095388725 | 0.7214683749683 | 0.664039871779 |
| Gausian NB | 0.786219789216 | 0.690232553468 | 0.4875167554554 | 0.485707150101 |

Two aspects influenced the performance of IDSX-Attention. We believe that SDAE has success to applied dimensional reduction with more representative features over our competitors using MI and PCA. The improvement algorithm using the stacking process becomes an essential factor in feature extraction in SDAE. We know that SDAE is a novel model of Auto Encoder for dimensional reduction

mechanism. The attention model also plays an important role in enhancing our model with sequence to sequence mechanism to get important value in the LSTM learning process.

Fig.6 is the confusion matrix evaluation result of IDSX-Attention and previous work based on LSTM and traditional model using Naïve Bayes. Naïve Bayes model (Fig. 6 (d), (e)) shows error detection near 50%. DoS-dominated error detection was detected as normal representation, and normal was detected as R2L. While the use of SDAE as dimensional reduction achieved better performance at 70%. It means the increasing performance by SDAE achieved above 25%. Applying SDAE is suitable to reduce miss detection of Naïve Bayes in the case of IDS application with tremendous achievement. Dimensional reduction using SDAE is responsible for removing unnecessary feature datasets.

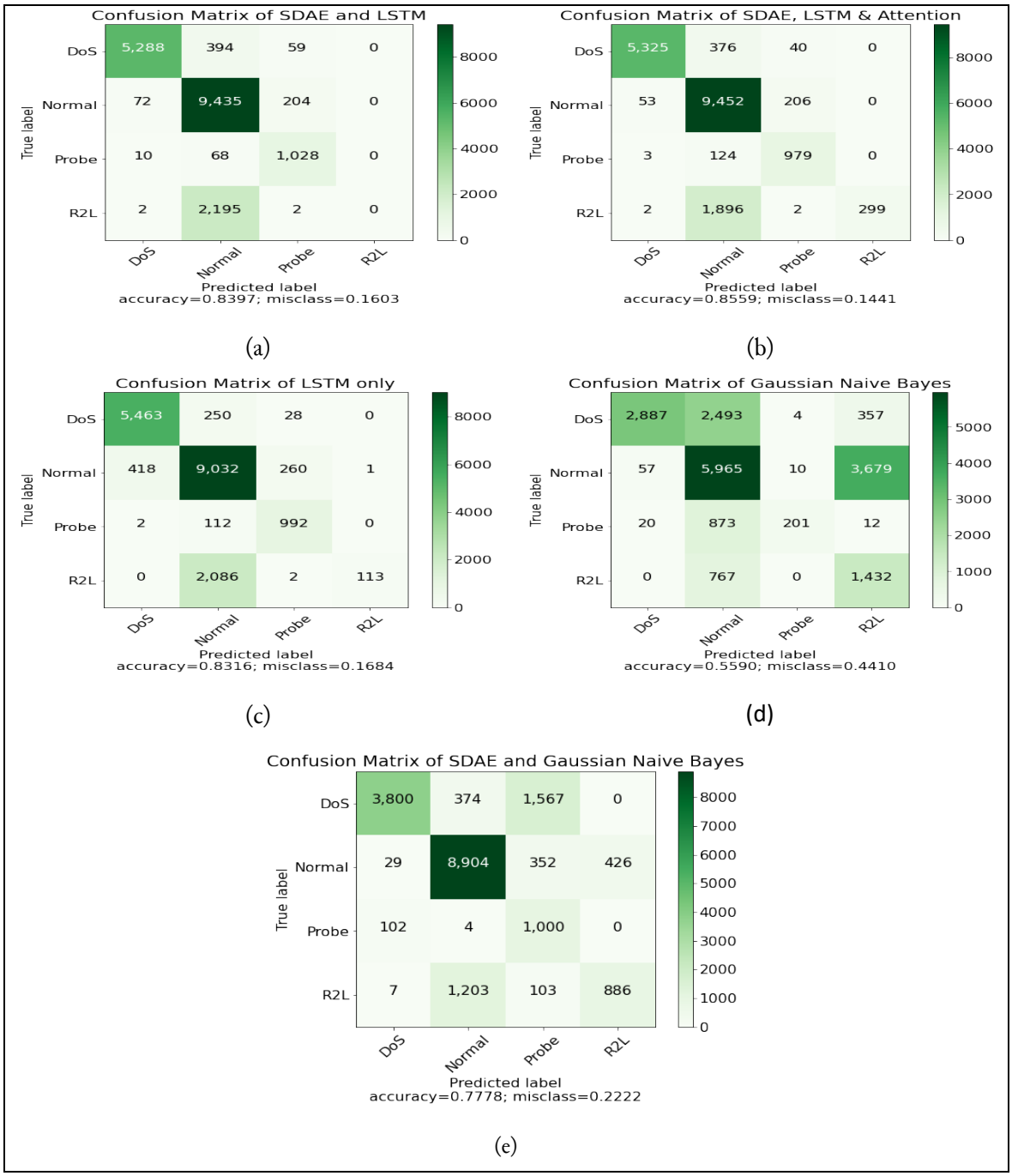


**Fig. 6.** Confusion matrix evaluation result

The second section of our experiment involves traditional LSTM without SDAE and Attention mechanism (see Fig. 6 (a), (b), (c)). LSTM achieved performance to reduce miss detection by 83.1% and error rate detection by 16.8%. In addition, LSTM influences significant achievement to increase true rate detection. The further section of our experiment considers adopting the Attention mechanism to enhance LSTM work for the classification task. According to the experiment report shown in Fig. 6 (b), the application of the Attention mechanism success in increasing effectiveness by 1.6%. The adoption of statistic pre-processing, SDAE as a dimensional reduction task, and LSTM with Attention mechanism play important roles in increasing IDS application performance.

Our experiment also considers applying the accuracy and losses test demonstrated in the illustration figure. The overall experiment is shown in Fig. 7. The number of epochs is 20 times for every training section for each classification model. The result of the evaluation is almost similar to the Confusion matrix test. Involvement of SDAE increases accuracy level and reduces loss (Fig. 7 (a, b, c,d, e).

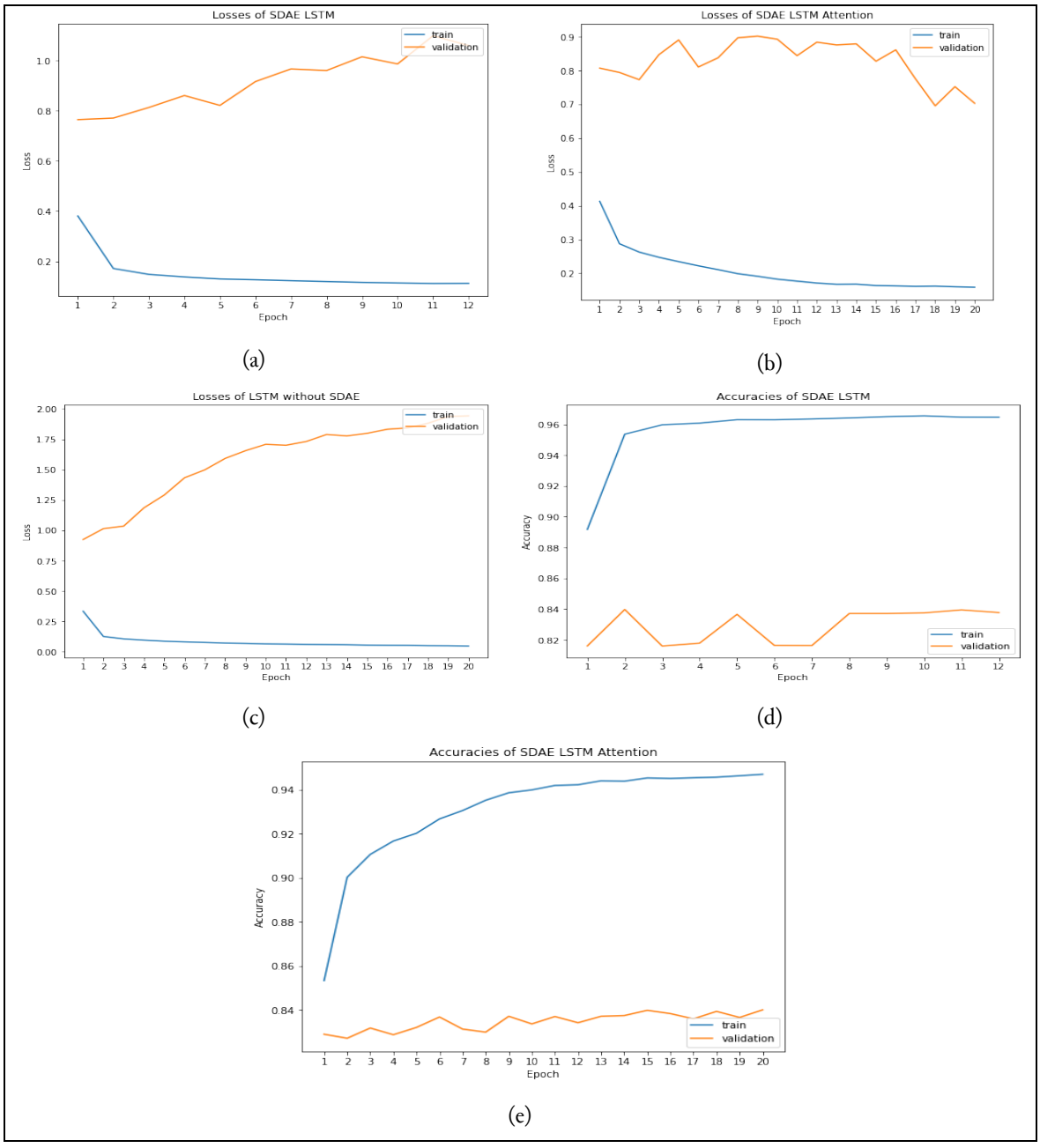


**Fig. 7.** Loss and Accuracy test

## 4. Conclusion

In this study, we aimed to improve the performance of Intrusion Detection Systems (IDS) by combining statistical methods MADE, dimensional reduction technique Stacked Denoising Autoencoder (SDAE), Long Short-Term Memory (LSTM), and Attention Mechanism. Our experimental results demonstrated that our proposed model improved the effectiveness of IDS detection. The performance of IDS is influenced by various factors, including the quality of datasets, the dimensional reduction method, and the algorithm used for the IDS classification task. We found that SDAE as a dimensional reduction technique successfully enhanced the performance of some machine learning algorithms compared to the common dimensional reduction approach using Principal Component Analysis (PCA) and Mutual Information (MI). As a result, applying SDAE improved the performance of LSTM task, which we believe is due to the dimensional reduction by SDAE. Our experimental results also showed that using SDAE significantly improved the performance of Naïve Bayes, LSTM, and LSTM-Attention. Furthermore, adopting the Attention mechanism in LSTM algorithm reduced error detection in the IDS classification task by more than 2% compared to the hybridization between SDAE and LSTM. The effectiveness of SDAE was due to the stacking process before computation into the autoencoder process. Additionally, the effectiveness of the Attention mechanism enhanced the LSTM's work by the sequence-to-sequence aspect that is owned by Attention feature. For future work, we suggest applying our proposed model to other datasets to validate its effectiveness further. Additionally, we recommend exploring the integration of other deep learning algorithms, such as Convolutional Neural Network with Attention (CNN-Attention), Generative Adversarial Network (GAN), and Graph Convolutional Neural Network (GCN), to further improve the performance of IDS.

### Acknowledgment

We thank Universitas Amikom Yogyakarta for research support and submission fee funding.

### Declarations

**Author contribution.** The first Author created a conceptual literature review, algorithm model, and collected datasets. The second Author is responsible for creating the article, developing the program, and data analysis. All the Authors discussed the result, analysis, and final manuscript.
**Funding statement.** The authors did not receive any research funding from any organization.
**Conflict of interest.** The authors declare no conflict of interest.
**Additional information.** No additional information is available for this paper.

### Data and Software Availability Statements

Data and Software availability statements provide a statement about where data and software supporting the results reported in a published article can be found, including hyperlinks to publicly archived datasets and software analyzed and generated during the study/experiments

### References

[1] A. Sunyoto and Hanafi, "Enhance Intrusion Detection (IDS) System Using Deep SDAE to Increase Effectiveness of Dimensional Reduction in Machine Learning and Deep Learning," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 4, p. 2022, doi: 10.22266/ijies2022.0831.13.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, no. 2, pp. 222–232, 1987, doi: 10.1109/TSE.1987.232894.

[3] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, "Deep Learning-Based IDSs," Springer, Singapore, 2018, pp. 35–45, doi: 10.1007/978-981-13-1444-5_5.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *J. Big Data*, vol. 8, no. 1, pp. 1–16, 2021, doi: 10.1186/s40537-021-00448-4.

[17] 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.

[18] 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.

[19] Hanafi, "Enhance Rating Prediction for E-commerce Recommender System Using Hybridization of SDAE, Attention Mechanism and Probabilistic Matrix Factorization," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 5, pp. 427–438, 2022, doi: 10.22266/ijies2022.1031.37.

[20] O. S. Shalom, H. Roitman, and P. Kouki, "Natural Language Processing for Recommender Systems," in *Recommender Systems Handbook*, New York, NY: Springer US, 2022, pp. 447–483, doi: 10.1007/978-1-0716-2197-4_12.

[21] P. Ren, Z. Ren, F. Sun, X. He, D. Yin, and M. de Rijke, "NLP4REC: The WSDM 2020 Workshop on Natural Language Processing for Recommendations," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, Jan. 2020, pp. 907–908, doi: 10.1145/3336191.3371884.

[22] 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.

[23] A. Trappey, C. V. Trappey, and A. Hsieh, "An intelligent patent recommender adopting machine learning approach for natural language processing: A case study for smart machinery technology mining," in

*Technological Forecasting and Social Change*, Mar. 2021, vol. 164, p. 120511, doi: 10.1016/j.techfore.2020.120511.

[24] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, "Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach," *Energies*, vol. 12, no. 17, p. 3310, Aug. 2019, doi: 10.3390/en12173310.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] S. Hochreiter, "Long Short-Term Memory," vol. 1780, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[30] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, pp. 577–585, 2015, doi: 10.48550/arXiv.1506.07503.