Scientific and Practical Cyber Security Journal (SPCSJ) 7(4): 36 – 51 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)

ENHANCING AUTOENCODER PERFORMANCE FOR INTRUSION DETECTION SYSTEMS VIA OPTIMAL BOTTLENECK SIZE OPTIMIZATION IN A TWO HIDDEN LAYER ARCHITECTURE

Seiba Alhassan[1,2], Gaddafi Abdul-Salaam*[1], Yaw Missah[1]

[1]Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
[2]Dr Hilla Limann Technical University

ABSTRACT. Sensitive data processed, stored, and transmitted on a computer requires a mechanism to protect it from unauthorized access. Several techniques have been proposed, including Intrusion Detection Systems (IDS), to protect computer networks from attacks. Autoencoders, a deep learning technique, have been explored by several researchers aiming to improve the performance of existing IDS. Despite the significant improvements seen with the use of autoencoders, the issues of low detection accuracy and high false alarm rates continue to be major problem. The architecture of a deep autoencoder, including the number of layers, neurons, and the bottleneck, affects its performance. This study is conducted to determine the optimal bottleneck size based on the architecture of a two-layer autoencoder. The study utilizes the benchmark dataset NSL-KDD to train, test, and validate the model. The experimental results from our proposed system reveal that the optimal bottleneck size for an autoencoder is obtained by setting it to 60% of the size of the previous hidden layer.

KEYWORDS. Autoencoder, IDS, encoder, decoder, bottleneck

INTRODUCTION

The benefits of computer networks have attracted various organizations, including healthcare, banks, educational institutions, security services, industry, transportation, hospitality, and individuals, to store sensitive information online. However, the increasing rate and sophistication of attacks on these networks pose a significant danger. Cybersecurity experts and academia have made considerable efforts to enhance cybersecurity. Although progress has been made, addressing the rising threat levels requires further attention. One extensively researched security technique is Intrusion Detection Systems (IDS). IDS, by their nature, facilitate early detection, enabling prompt actions to mitigate attack severity. According (Xu et al. 2021), IDS's ultimate goal is to classify network traffic as normal or malicious. These systems are built using machine learning and deep learning techniques and can be categorized as anomaly-based or signature-based IDS.

Signature-based IDS maintains a database of known attacks, comparing incoming network traffic against this database. Anomaly-based IDS, conversely, establishes a normal profile and flags incoming traffic deviating from this profile as an attack. Both approaches have strengths and weaknesses. For instance, anomaly-based systems are prone to high false alarms but can detect novel attacks. Signature-based IDS struggle to identify new attacks and require frequent database updates, making them computationally expensive, but they have lower false alarm rates.

Another classification criterion for IDS is their implementation location. Network Intrusion Detection Systems operate at the network level, monitoring data packets and classifying them as normal or malicious. Host-based IDS involves installing software on individual systems for tracking purposes.

While various researchers have achieved substantial success with IDS techniques, the accuracy of intrusion detection remains a significant research challenge. (Alam and Ahmed 2023; Logeswari, Bose, and Anitha 2023; Kasongo 2023; Shukla and Kumar 2023; Ramasamy and Eric 2023; Pranto et al.

2022; Makarand 2022; Hendi, Verawati, and Hardi 2022; Das 2022; Li et al. 2022; Garg, Kumar, and Shyamasundar n.d.) have employed machine learning algorithms for IDS implementation, reporting impressive results. However, machine learning has limitations, as confirmed by (Shone et al. 2018), who emphasized the need for human expert interaction. To address this limitation, researchers are turning to deep learning techniques, such as Autoencoders, which have shown promise in IDS research. Several researchers (Schmidt 2020; Y. Song, Hyun, and Cheong 2021; Haripriya and Jagadeesh 2022; Sabir, Ahmad, and Alghazzawi 2023; Almaiah et al. 2022; Shahid et al. 2019; Siddique et al. 2019; Wang et al. 2022; Yu, Long, and Cai 2017; Zhang, Yu, and Li 2018)have recently explored Autoencoders and reported impressive performance.

Autoencoders, as a deep learning technique, consist of three main components: the input, which comprises the dataset; the encoder, which transforms high-dimensional data into a lower-dimensional space; and the decoder, which converts the lower-dimensional space back to the output. The output is exists. Figure 1 illustrates a standard autoencoder with two hidden layers.

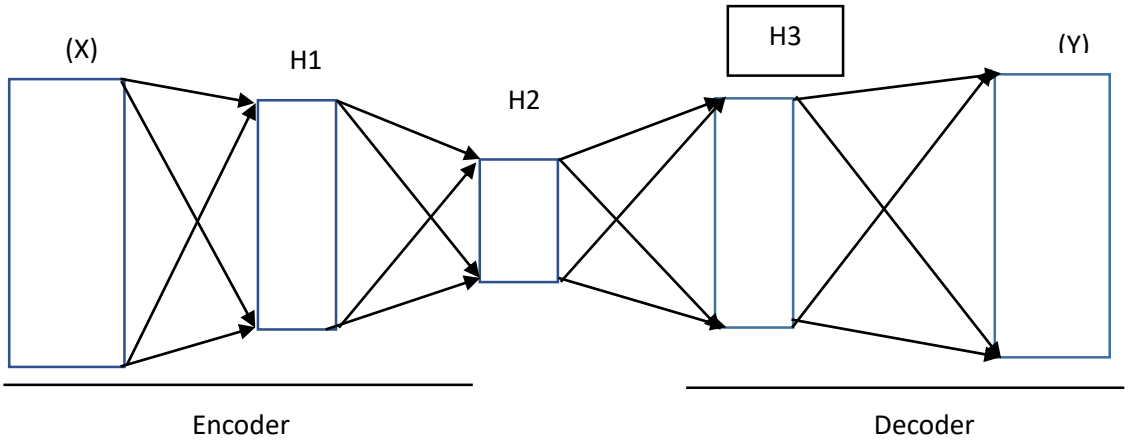

Figure 1: Autoencoder Architecture

Figure 1 illustrates that, when given an input of size X which is compressed into a lower dimension of size H2 (where X > H2), the bottleneck is then converted to Y, which is approximately the size of X. In anomaly detection systems like IDS, the autoencoder is typically fed with normal data, and a threshold value is established. Subsequently, when the model is fed input containing both attacks and normal data, any deviation from the threshold value is considered an abnormality or an attack. This property makes autoencoders suitable for detecting zero-day attacks. However, the full potential of autoencoders is not fully realized due to the lack of a generally accepted standard architecture for the latent space or bottleneck, resulting in lower detection accuracy.

(Y. Song, Hyun, and Cheong 2021) conducted a study aimed at analyzing the impact of the dimension of the latent space on the model's performance. However, their study did not identify the optimal latent size that would lead to higher model performance. This absence of a suggested optimal latent size often results in a trial-and-error approach, which is time-consuming and delays the practical implementation of deep autoencoders.

The contributions of this study include:

1. Designing and implementing various latent space sizes for a two-hidden-layer autoencoder.
2. Suggesting an optimal latent space size to expedite the practical implementation of autoencoders for IDS.

The rest of this work is divided into four main sections. Section 2 reviews related literature. Section 3 outlines the methodology used to implement the proposed system. Section 4 covers the results and discussion. The final section presents the conclusion.

2.0 Literature Review

(Mirsky et al. 2018) conducted a study introducing Kitsune, a neural network-based Network Intrusion Detection System (NIDS) designed for efficiency and plug-and-play deployment. Kitsune achieves this through efficient tracking of network behavior across channels and utilizes an ensemble of autoencoders known as KitNET for anomaly detection. The study focuses on the online machine learning process of the framework and evaluates its performance in terms of detection accuracy and runtime efficiency.

The authors highlight that KitNET, an online algorithm, exhibits competitive performance comparable to batch or offline algorithms and, in some cases, outperforms them. Notably, the algorithm's efficiency is demonstrated by its ability to operate on a single core of a Raspberry Pi device, with potential for even stronger CPUs.

(T. Song et al. 2019)presented a study that introduces the LSE-VAE (Latent Space Encoding Variational Autoencoder) model as an innovative approach to sentence generation. By incorporating distinct prior latent distributions tailored to different sentences and structuring the latent space based on sentence similarity, the model effectively captures a substantial and informative latent representation. The research evaluates the LSE-VAE's performance through a combination of automated metrics and empirical analysis.

In comparison to the conventional Variational Autoencoder (VAE), the LSE-VAE exhibits superior reconstruction capabilities, generating sentences of higher quality and greater diversity. Notably, the latent space learned by the LSE-VAE maintains the desirable attributes of continuity and smoothness observed in VAE-based latent spaces while further excelling at distinguishing sentences with varying degrees of similarity. An intriguing aspect of the LSE-VAE model is its enhanced ease of training, requiring fewer complex engineering strategies such as KL cost annealing. The determination of hyperparameters is streamlined through analytical derivation, taking into account factors such as latent variable dimensions and modeling requirements. This analytical approach contributes to the model's practicality and ease of implementation.

In connection to the previous literature review, where Kitsune was introduced as a neural network-based NIDS, both studies contribute to advancing their respective fields through innovative modeling approaches. Just as Kitsune enhances intrusion detection through efficient autoencoder ensembles, the LSE-VAE model elevates sentence generation with a specialized latent space arrangement. The intersection of neural network methodologies across diverse domains underscores the versatility and impact of deep learning techniques in addressing complex challenges.

(Sindian and Sindian 2020) also presented a study introducing a novel approach called the Deep Sparse Autoencoder-based Approach with two hidden layers (EDSA) for feature extraction and DDoS attack detection. The core objective of this research is to leverage autoencoders to extract representative features from the CICDDoS2019 dataset, subsequently minimizing classification errors and enhancing the accuracy of DDoS attack detection.

The empirical analysis conducted on the proposed EDSA technique demonstrates its remarkable performance in terms of detection accuracy. A significant improvement is observed when compared to other network models across various performance indicators, including accuracy, detection rate, precision, and specificity. Notably, the false positive rate is considerably reduced, underscoring the effectiveness of the EDSA method. For the CICDDoS2019 dataset, the proposed technique achieves an impressive 98 percent detection accuracy and a minimal 1.4 percent false positive rate. Their

study's findings suggest the potential for further enhancements and exploration. The authors propose the incorporation of recent computer algorithms like K-means clustering, potentially introducing additional layers within the Sparse Autoencoder (SAE) structure to further reduce feature dimensions. Furthermore, the study envisions the application of alternative classification algorithms beyond the scope of the current research.

In (Sindian and Sindian 2020), a study is proposed autoencoders as a powerful tool for capturing underlying factors in various types of datasets. Autoencoders' latent representations have been extensively studied in the context of facilitating interpolation between data points by decoding convex combinations of latent vectors. However, this interpolation process often results in artifacts or unrealistic outcomes during the reconstruction phase. The authors contend that these discrepancies arise from the structure of the latent space and the inherent deviation of naively interpolated latent vectors from the actual data manifold.

In response to these challenges, the paper introduces an innovative regularization technique aimed at reshaping the latent representation. This regularization strategy strives to align the latent manifold with the training images, ensuring consistency and fidelity. Moreover, the technique promotes smoothness and local convexity within the manifold, addressing the issues associated with interpolation artifacts and unrealistic outcomes.

The proposed regularization technique not only facilitates accurate interpolation between data points, as evidenced in the study, but also serves as a versatile approach to combat overfitting. Furthermore, it offers the potential to generate new samples for data augmentation, showcasing its broader applicability in enhancing dataset diversity and model generalization.

This research contributes to the field of autoencoders by addressing a critical concern in latent space interpolation. By refining the latent manifold's structure, their study presents a robust solution that advances the quality and realism of interpolation results. Additionally, the regularization technique's versatility in preventing overfitting and generating augmented data underscores its practicality and significance in diverse machine learning applications.

(Xu et al. 2021) introduced a study that presents a novel 5-layer autoencoder (AE)-based model designed to enhance the detection of anomalous network traffic. The development of this model is informed by a thorough and meticulous examination of key performance indicators and their impact on detection accuracy within an AE framework. Through a rigorous evaluation, the authors establish that the proposed 5-layer architecture, combined with an innovative data pre-processing methodology and specific loss metrics, yields optimal results in terms of accuracy and detection precision.

Central to the success of the proposed model is the use of Mean Absolute Error (MAE) as the basis for the reconstruction loss function. The authors highlight how this choice of loss metric contributes to improved accuracy in network anomaly detection.

The optimized 5-layer architecture, with carefully determined numbers of neurons in hidden and latent layers, outperforms alternative model architectures. The evaluation is conducted on the NSL-KDD dataset, where the proposed model achieves impressive performance metrics, including accuracy, precision, recall, and F1-score.

Their study acknowledges the adaptability of the model beyond the specific dataset used for training. While currently focused on NSL-KDD, the proposed model demonstrates an ability to recognize abnormal network traffic patterns effectively. Future plans include testing the model's generalizability across different intrusion attack samples and datasets from diverse applications, such as Android-based malware and medical annotations. The authors also express a commitment

to expanding their work to encompass multi-class classification and assessing the model's performance in real-world operational network environments.

(Y. Song, Hyun, and Cheong 2021) explored the domain of intelligent Network Intrusion Detection Systems (NIDS) and their application of deep learning techniques to counteract the evolving landscape of network attacks. The focus is on leveraging autoencoders as a means to effectively identify new attack patterns and mitigate the challenges posed by the labor-intensive labeling of data. While autoencoders prove adept at detecting unknown attack types, the process of fine-tuning model architecture and hyperparameters to achieve optimal detection performance can be a resource-intensive endeavor, potentially hindering the practical implementation of autoencoder-based NIDS.

To address this challenge, the study takes a rigorous approach by investigating autoencoders using established benchmark datasets, including NSL-KDD, IoTID20, and N-BaIoT. The research systematically explores multiple combinations of model structures and latent sizes within a simple autoencoder framework. Through this thorough evaluation, the article sheds light on the critical role that the latent size of an autoencoder model plays in influencing the performance of an Intrusion Detection System (IDS).

(Xu et al. 2021)delves into the challenges posed by the emerging paradigm of the Internet of Things (IoT), which, while offering numerous benefits, is susceptible to cyberattacks due to its resource-constrained and heterogeneous nature. Successful network intrusions in IoT networks can have far-reaching consequences, compromising valuable consumer and industry information. To counteract these security challenges, the article introduces a novel approach: a lightweight deep autoencoder (DAE)-based cyberattack detection framework.

The efficacy of the proposed framework is substantiated through evaluation using two standard and open-source datasets: NSL-KDD and UNSW-NB15. In both binary class and multi-class scenarios, the proposed DAE achieves impressive accuracies, attaining 98.86% and 98.26% for NSL-KDD, as well as 99.32% and 98.79% for the UNSW-NB15 dataset.

To establish the robustness of the approach, the article compares the performance of the proposed attack detection framework with several state-of-the-art intrusion detection schemes. The experimental results underscore the promising nature of the proposed scheme in effectively detecting cyberattacks within IoT networks.

The concept of latent space and architecture serves as a fundamental thread connecting the reviewed articles. Latent space refers to a compressed and abstract representation of data that captures underlying patterns and features. Architecture, on the other hand, refers to the design and structure of neural networks used to model and manipulate data.

Both (Y. Song, Hyun, and Cheong 2021; Xu et al. 2021) underscore the importance of optimizing neural network architectures to achieve desired outcomes. (Y. Song, Hyun, and Cheong 2021)focus on autoencoder-based NIDS, emphasizing the need for careful architecture design to achieve optimal intrusion detection performance. Similarly, (Xu et al. 2021) meticulously explore various architectural configurations to develop an effective model for detecting anomalous network traffic. In both cases, the architecture's structure and design choices influence the characteristics of the latent space, which, in turn, impacts the model's performance.

In summary, latent space and architecture are central concepts that interplay across the reviewed articles. Whether in the context of anomaly detection, sentence generation, model optimization, or regularization, the design choices made in constructing neural network architectures directly impact the nature and quality of the latent space representation, ultimately influencing the effectiveness and performance of the models in their respective domains.

The perspectives of these authors suggest that much more attention needs to be paid to the issue of latent space to achieve optimal performance of IDS. In view of this, the next section of this study will clearly outline the processes, procedures, and tools necessary to implement a study aimed at obtaining an optimal latent space that will consistently guarantee impressive performance of an autoencoder. These findings will help improve the detection accuracy of current and existing IDS.

3.0 Methodology

3. 1 Autoencoder

The model designed for this study is the autoencoder for Network Intrusion Detection. The autoencoder is a deep learning algorithm that takes input data (X) and compresses it to a lower dimension known as the bottleneck (B) in a process known as encoding. The bottleneck is then used to reconstruct the output (Y) in a process known as decoding.

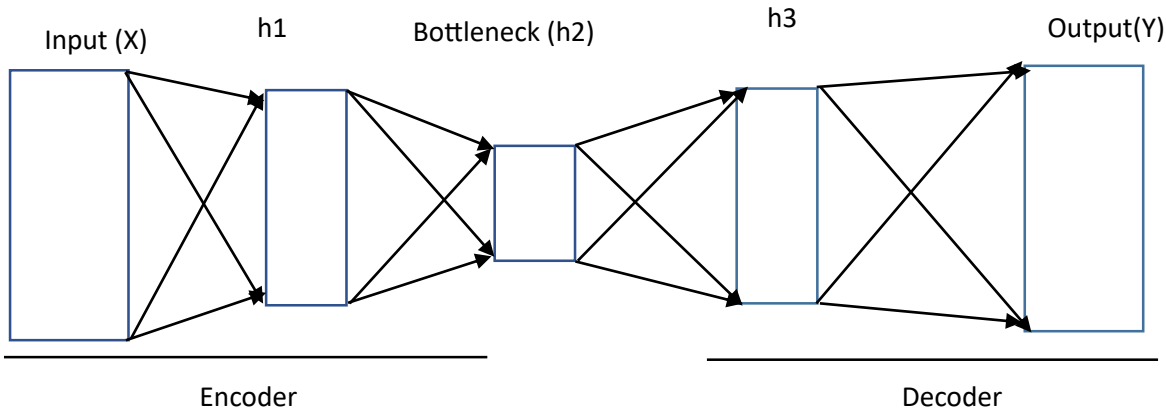

Figure 2: Autoencoder model

X: Input (input features)

Y: Reconstructed output data

h2: bottleneck

f: activation function for the encoder

g: activation function for the decoder

The autoencoder goal is to learn a mapping X to Y such that the reconstructed output Y is as close as possible to the input data X. The architecture consists of an encoder and a decoder

i. Encoder

The encoder function takes the input X and maps it to hidden representation h2 via two hidden layers:

$h1 = f(W1.X+b1)$ ……... (1)

$h2 = f(W2.X+b2)$ …… (2)

Where:

W1 represent the weight of the first hidden layer

b1 represent the bias of the first hidden layer

W2 represent the weight of the Second hidden layer

B2 represent the weight of the second hidden layer

The decoder function maps the bottleneck h2 to the reconstructed output Y

Via two hidden layers:

$H3 = f(W2.h2 + b2)$ ……..... (3)

$Y = g(W2.\ h3 + b2)$ …….....(4)

Where:

W2: Weights of the first hidden layer of the decoder

B2: bias of the first hidden layer of the decoder

W2: Weights of the second hidden layer of the decoder

B2: bias of the second hidden layer of the decoder

Loss Function

This study made use of the mean square error (MSE to measure the difference between the input X and the reconstructed output Y and this is represented mathematically as

$MSE\ (X, Y) = \frac{1}{n}\sum_{i=1}^{n}(X_i\text{-}Y_i)$ …… (5)

3.1: Our propose System

The primary objective of this study is to investigate how the architecture of an autoencoder influences the performance of an Intrusion Detection System (IDS). Specifically, the study focuses on determining the optimal bottleneck size that leads to improved IDS accuracy. The study employs a two-hidden-layer autoencoder for its investigation. The research plan involves conducting two experimental setups to achieve this goal.

In the first experimental setup, a constant number of 50 neurons is maintained in the first hidden layer, while the last hidden layer (bottleneck) is varied. The values considered for the bottleneck size include 40, 30, 20, and 10. The outcomes of this setup will provide insights into the impact of varying bottleneck sizes on IDS accuracy and guide the subsequent steps in the investigation.

It is worth noting that this study introduces a unique approach distinct from prior research on the same topic. Previous studies have explored whether the bottleneck size influences model performance but have not delved deeper to ascertain the optimal bottleneck size for achieving superior model performance. For instance, a study by Song, Hyun, and Cheong (2021) conducted a similar investigation but primarily focused on assessing the effect of the bottleneck size on model performance.

In summary, this study aims to contribute to the existing body of knowledge by not only examining the influence of bottleneck size on IDS model performance but also determining the precise bottleneck size that leads to optimal performance. This refined approach will provide valuable insights into designing more effective autoencoder architectures for intrusion detection.

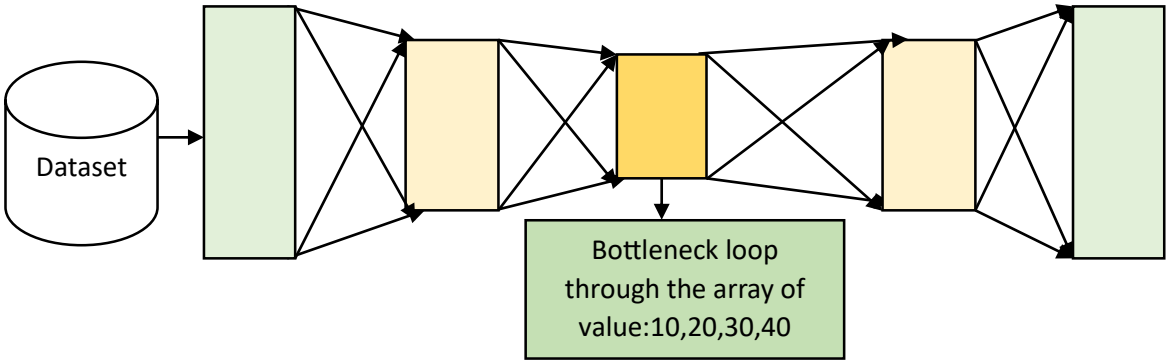

Figure 3: Preliminary setup to obtain optimal bottleneck value

In the Figure 3 above, the detection accuracy is recorded for each value in the array, and the value that yields the highest accuracy is chosen. This selected value is subsequently utilized as a seed value to conduct the next experimental setup, aiming to determine the suggested optimal bottleneck value for a two-hidden-layer autoencoder designed for Network Intrusion Detection. The next setup involves obtaining three values, represented as X ± 5, as illustrated in the Figure below.

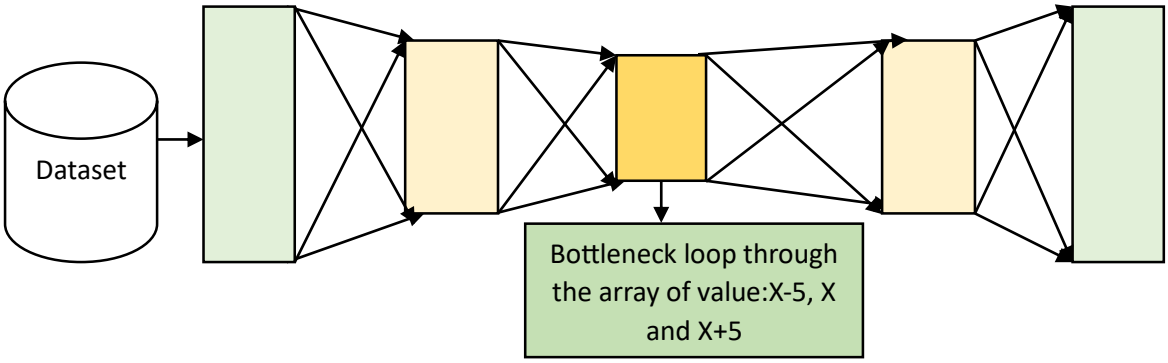

Figure 4: Setup to obtain optimal bottleneck value

The first component for our proposed system is the dataset. These datasets are used for training and testing the proposed system. The datasets include the CIC-IDS2017, and NSL-KDD. The sections below take a detail look at each of them.

3.3 Datasets

3.3.1 NSL-KDD dataset

The NSL-KDD dataset is a well-known IDS dataset that is extensively employed by numerous researchers to validate their models. Its frequent utilization simplifies the process of comparing research outcomes with those of prior studies. According to (Tavallaee et al., 2009), NSL-KDD was developed to address the inherent issues associated with the KDDCup99 dataset. NSL-KDD remains relevant in contemporary research due to its capacity to yield consistent results, facilitating effective comparisons with other studies. This advantage stems from the dataset's balanced distribution of training and testing records, allowing for the entire dataset to be used without necessitating the selection of a randomly chosen subset. NSL-KDD encompasses four attack classes and a normal class. The instance counts per

class are presented in Table 1, while Table 2 provides an overview of the attack types and categories present within the NSL-KDD dataset.

Table 1:Classes and number of instances in NSL-KDD dataset

S/N	CLASS	NO OF INSTANCE
1	R2L	52(0.04%)
2	U2R	995(0.78%)
3	probe	11656(9.25%)
4	DoS	45927(36.47%)
5	Normal	67343(53.46%)

Table 2: Types and categories of attacks in NSL-KDD dataset

TYPE OF ATTACK	CATEGORIES OF ATTACK
Probe	N map, Portsweep, Satan, saint(6), Mscan
DoS	Worm (10), Back, Land Neptune, Process table, Udpstorm,Pod, Teardrop, Smurf, Apache 2.
U2R	Load Module,Perl, Sql attack, Buffer_overflow, Rotkit,Xterm ,Ps(7)
R2L	Spy, Xlock,Guess_Password, Ftp_write, Httptunnnel, Named(16), Pht, Multihop, Ftp_write, Send fmail, Name(16), Xsnoop, Waremaster, Snmp guesss, Snmp getattack

3.4 Data preprocessing

3.4.1 One hot encoding

The proposed systems AE-LSTM cannot directly process NSL-KDD, dataset in its original form. The one-hot- encoding is used to transform the non-numeric features into numeric feature for the AE to process it. NSL-KDD dataset has 38 numeric features and three non-numeric features. The nonnumeric feature such “protocol-type”, “flag” and “service” need to be converted into numeric format.

i. The first non-numeric feature to be converted into numeric feature is the protocol-type. The protocol type has three main attributes namely the ‘TCP’,’UDP’ and ‘ICMP’ which are encoded as follows as shown in Table 3.

Table 3: converting non-numeric features to numeric features

Protocol-type	TCP	UDP	ICMP
encoded	1,0,0	0,1,0	0,0,1

Next, the “flag” and “service features” are converted into numeric features. The service feature has seventy (70) different attributes and so by using the same method in the step (i) above each attribute of service feature is mapped to 70 distinct binary attributes. Similarly, the flag feature also has 11 different attributes and is also converted into numeric features by mapping it to 11 distinct binary attributes. As result of this transformation, the 41 features of NSL-KDD dataset are transformed into 112 distinct features.

3.4.2 Normalization

The datasets are first normalized to enhance the performance and reliability of our model by converting all numeric columns to a common scale. The equation three (3) below shows how the min-max technique is used to perform the normalization task.

y = x-min/max-min……..…(6)

Were

y = new value of each entry

Min = minimum value for each data points

Max = maximum value for each data points

A similar process is also used to prepare the CIC-IDS2017 dataset for the autoencoder learning algorithm.

3.4.5 Data Splitting

The data that has been transformed is then split into the ratio 75:25 for training and testing respectively.

3. 5 Metrics of evaluation

Intrusion detection systems performance is evaluated based on a number of metrics including the accuracy, precision, F1-score and Recall. The others are:

True positive: correctly classified attacks in a data sample

True Negative: Normal traffic in a data sample that has been correctly classified as Normal

False positive: Normal traffic in data sample wrongly classified as an attack

False negative: Malicious traffic in a data sample that has been wrongly classified as Normal

The metrics are calculated as follows:

Accuracy measures: the total number of data samples correctly classified as true positive or true negative. Higher accuracy for the balanced dataset is an indication of good performance. The Equation 7 below shows how accuracy is calculated.

Accuracy (ACC) = $\frac{TP+TN}{TP+TN+FP+FN}$ …… (7)

Recall which is also called true positive rate is the proportion of correctly predicted positive instances of a class to the overall instance of the same class. A higher recall rate that ranges from 0 to 1 indicates a better model performance. The Equation 8 below shows how the Recall is calculated

Recall = $\frac{TP}{TP+FN}$ …… (8)

Precision is the ratio of positive instances that are correctly predicted to the ratio of all predicted samples for a class. Recall and Precision are always paired when evaluating model performance. The Equation 9 below shows how the precision is calculated

Precision = $\frac{TP}{TP+FP}$ …… (9)

F1-score is computed by taking the harmonic mean of precision and recall. F1-score normally calculates the tradeoff between precision and recall. F1-score is calculated as shown in equation 8 below

$$\textbf{F1-score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad \text{..... (10)}$$

3.6 Experimental setup

The experimental results were obtained using the following specifications to construct the model in Python on the Google Colab platform, utilizing a CPU processor. The number of epochs was set to 100, and the batch size was set to 500. For the activation function, the ReLU activation was employed for both the encoding function and the hidden layers of the decoding function. Subsequently, the softmax function was utilized as the output function.

The experimental setups were executed with a two-hidden-layer architecture, where the first layer was kept at a constant size of 50 units. In each setup, the latent space was varied using array elements 10, 20, 30, and 40, maintaining the ratio 50:10, 50:20, 50:30, and 50:40, respectively. A distinct model was built for each configuration, and the corresponding results were recorded. Notably, to ensure consistent results, each setup was executed only once, thereby preventing interference from previous runs.

The primary objective of these experiments was to determine the optimal latent space size that leads to the highest accuracy for intrusion detection. This optimal latent space size, denoted as X, was identified. To further refine the architecture, latent space sizes of X-5, X, and X+5 were generated. These new configurations were then explored to derive the most optimal bottleneck size, aiming to enhance the performance of the intrusion detection system using a deep autoencoder.

This investigation targeted two benchmark datasets, namely NSL_KDD and CIC-IDS2017. By varying the latent space size and analyzing the resulting accuracy, the goal was to pinpoint the most suitable position within the array of elements. This "best" latent space size, represented by X, served as a foundation for subsequent analyses to fine-tune the architecture for improved intrusion detection capabilities.

4.0 Results and discussion

4.1 Preliminary Results and Latent Space Correlation:

The preliminary experimental results (Table 4 and Table 5) provide an initial glimpse into the impact of different latent space sizes on intrusion detection system performance. Notably, latent space size 30 consistently emerges as a high-performing configuration across both the NSL-KDD and CID-IDS2017 datasets. This observation is intriguing, as it aligns with the previously proposed hypothesis: the optimal latent space size should be around 60% of the preceding hidden layer's size. This alignment hints at the potential validity of this latent space correlation. Figure 5 and Figure 6 below show the pictorial view of the results from the preliminary study.

Table 4: Result of Preliminary experimental for NSL-KDD dataset

Model	Accuracy	TPR	FPR	Precision	Recall	F1-Score
50,10,50	84.86%	76.0%	0.065	92.12%	85.13%	88.48%
50,20,50	86.74%	85.2%	0.070	92.20%	87.22%	89.64%
50,30,50	91.75%	88.0%	0.073	92.55%	97.97.75%	95.55%
50,40,50	75.32%	60.7%	0.060	91.39%	81.00%	85.88%

Table 5: Result of Preliminary experimental for CID-IDS2017 dataset

Model	Accuracy	TPR	FPR	Precision	Recall	F1-Score

50,10,50	90.77 %	89.00%	0.065	92.5714%	93.13%	0.928499
50,20,50	92.88%	93.50%	0.070	93.0348%	96.22%	0.946006
50,30,50	95.98%	98.97%	0.068	93.5710%	98.85%	0.964679
50,35,50	83.10%	87.02	0.060	92.5267%	88.00%	0.902066

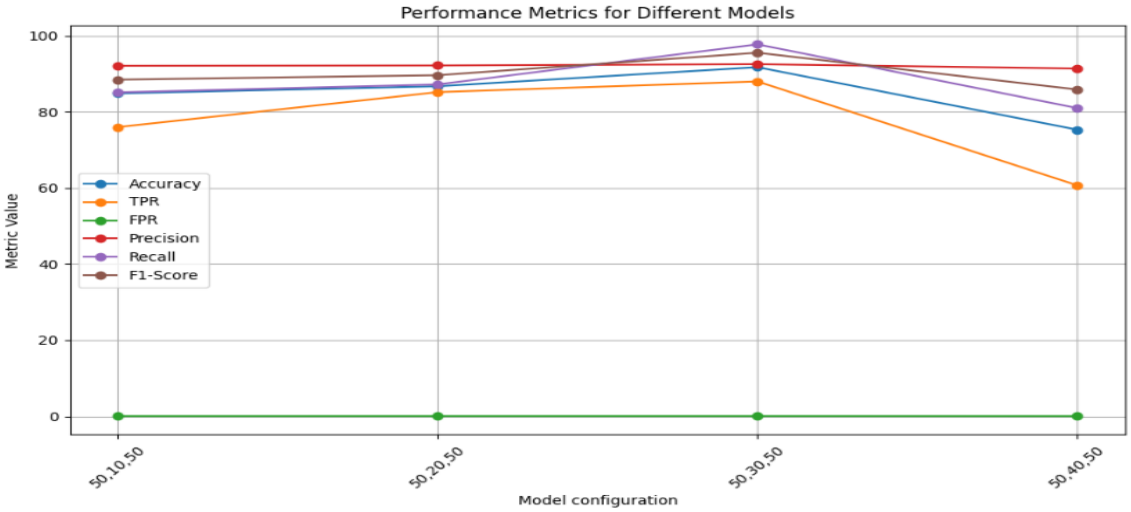

Figure 5: Bottlenecks vr metrics of evaluation for NSL-KDD dataset in preliminary study

Figure 6: Bottlenecks vr metrics of evaluation for CIC-IDS2017 dataset in preliminary study

4.2 Final results for NSL-KDD Dataset

In the context of the NSL-KDD dataset, the results reveal intriguing trends. Latent space size 30 emerges as a configuration that consistently maintains high accuracy, TPR, and F1-score values. The latent space correlation's manifestation in the final results bolsters its validity and utility in architecting effective intrusion detection systems. The Figure 7 below show clearly the various model configurations(bottlenecks) and their performance for our final study using NSL-KDD dataset.

Table 6: Results for final Experimental study using NSL-KDD dataset

Model	Accuracy	TPR	FPR	Precision	Recall	F1-Score
50,25,50	88.02%	77.00%	0.067	91.9952%	94.34%	90.0352%
50,30,50	91.75%	88.00%	0.073	92.3400%	97.97%	95.55%

50,35,50	89.66%	76.00%	0.073	91.2365%	95.99%	91.2533

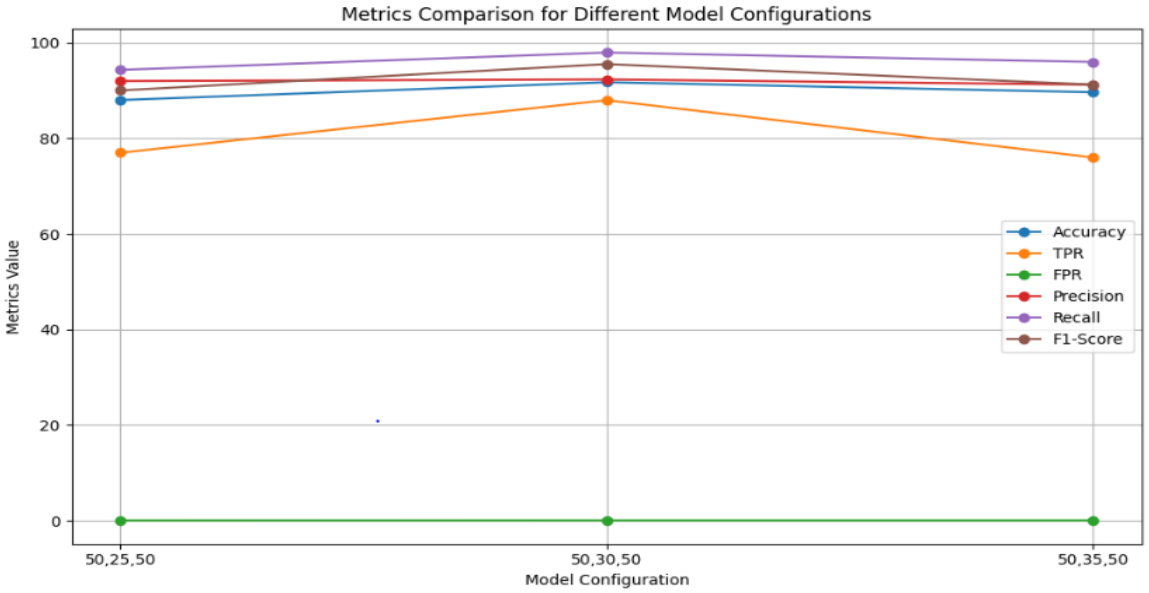

Figure 7: Bottlenecks vr metrics of evaluation for NSL-KDD dataset in Final study

4.3 Final results for CIC-IDS2017 Dataset

The findings from the CIC-IDS2017 dataset further substantiate the significance of the latent space correlation. Latent space size 30 continues to exhibit exceptional performance, reflecting its potential as a universal configuration guideline. The high TPR and F1-score values validate its effectiveness in detecting true anomalies while maintaining a balance against false positives. Figure below provides the pictorial representation for the various bottlenecks' performances for our final study using CIC-IDS2017.

Table 7: Results for final Experimental study using CIC-IDS2017 dataset

Model	Accuracy	TPR	FPR	Precision	Recall	F1-Score
50,25,50	93.50%	0.92.66	0.087	0.920245	95.01%	0.934934
50,30,50	95.98%	0.9897	0.068	0.93571	98.85%	0.961381
50,35,50	94.12%	0.9426	0.091	0.917623	96.75%	0.941902

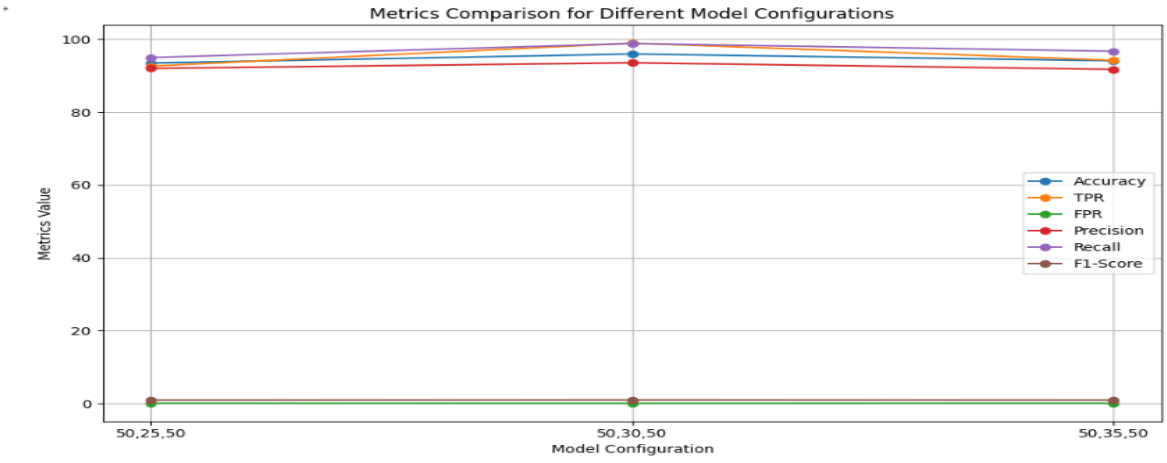

Figure 8: Bottlenecks vr metrics of evaluation for CIC-IDS2017 dataset in final study

4.4 Discussion

4.4.1 Impact on Intrusion Detection Systems:

The consistent success of latent space size 30 in both datasets underscores its effectiveness in boosting the performance of intrusion detection systems. This result holds practical implications for system architects and cybersecurity practitioners. It provides them with a concrete benchmark to guide architecture design, ensuring enhanced accuracy and precision in detecting intrusions.

The latent space correlation, where the optimal latent space size is approximately 60% of the preceding hidden layer's size, serves as a pivotal takeaway from this research. This empirical observation bridges the gap between theoretical understanding and practical application, offering a valuable heuristic for system designers aiming to optimize autoencoder-based intrusion detection systems.

The study's significance lies in its embodiment of the synergy between AI and security. By rigorously testing and validating latent space configurations, this research demonstrates how AI techniques can be harnessed to address pressing security challenges. The results showcase how AI-driven insights translate into actionable strategies for enhancing cybersecurity measures.

While this study illuminates the latent space correlation's potential, future research could explore its applicability to a broader range of datasets and intrusion scenarios. Additionally, investigating more intricate autoencoder architectures and considering other neural network techniques could uncover further optimization opportunities and contribute to the evolution of intrusion detection systems.

Armed with the findings, practitioners can confidently implement autoencoder architectures with latent space sizes around 60% of the preceding hidden layer's size. This practical application of research contributes directly to improving the robustness and efficiency of intrusion detection systems in real-world scenarios.

5.0 Conclusion

Finally, the constant success with a latent space size of 30 found in both datasets highlights its effectiveness in improving intrusion detection system performance, offering system architects and cybersecurity practitioners a concrete benchmark. The optimal size of the identified latent space correlation is about 60% of the size of the previous hidden layer. This is an important finding that connects the theoretical and practical domains and provides a useful heuristic for system designers who want to optimize autoencoder-based intrusion detection systems. The study shows how AI techniques, through thorough testing and validation of latent space configurations, may address urgent security concerns and convert into beneficial outcomes. It also represents the successful synergy between AI and security. practical methods for strengthening cybersecurity defenses. While shedding light on the latent space correlation's potential, further research opportunities include investigating how well it applies to various datasets and intrusion scenarios, investigating more complex autoencoder architectures, and taking into account alternative neural network techniques to find even more optimization opportunities for the development of intrusion detection systems. Equipped with these discoveries, professionals can safely execute autoencoder structures with latent spaces around 60% larger than the previous hidden layer's size, so directly enhancing the resilience and effectiveness of intrusion detection systems in practical scenarios.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Competing Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability:

The dataset NSL-KDD is publicly available on: https://www.unb.ca/cic/datasets/nsl.html

The dataset CIC-IDS2017 is publicly available on: https://www.unb.ca/cic/datasets/ids-2017.html

REFERENCES

1. Alam, Naushad, and Muqeem Ahmed. 2023. "Zero-Day Network Intrusion Detection Using Machine Learning Approach," no. April: 194–201.

2. Almaiah, Mohammed Amin, Omar Almomani, Adeeb Alsaaidah, Shaha Al-otaibi, Nabeel Bani-hani, Ahmad K Al Hwaitat, Ali Al-zahrani, Abdalwali Lutfi, Ali Bani Awad, and Theyazn H H Aldhyani. 2022. "Machine Kernels."

3. Das, Abhijit. 2022. "An Efficient Feature Selection Approach for Intrusion Detection System Using Decision Tree" 13 (2).

4. Garg, Deepak, N. V. Narendra Kumar, and Rudrapatna Shyamasundar. n.d. "Information Systems Security : 15th International Conference, ICISS 2019, Hyderabad, India, December 16-20, 2019, Proceedings," 345. Accessed February 7, 2022. https://www.kobo.com/us/en/ebook/information-systems-security-4.

5. Haripriya, C, and M P Prabhudev Jagadeesh. 2022. "An Efficient Autoencoder-Based Deep Learning Technique to Detect Network Intrusions" 13 (7): 1–10. https://doi.org/10.14456/ITJEMAST.2022.142.

6. Hendi, Alva, Ike Verawati, and Richki Hardi. 2022. "An Intrusion Detection System Using SDAE to Enhance Dimensional Reduction in Machine Learning" 6 (June): 306–16.

7. Kasongo, Sydney Mambwe. 2023. "A Deep Learning Technique for Intrusion Detection System Using a Recurrent Neural Networks Based Framework." *Computer Communications* 199 (October 2022): 113–25. https://doi.org/10.1016/j.comcom.2022.12.010.

8. Li, Yue, Ang Li, Anxing Wen, and Xian Xie. 2022. "Research on Intrusion Detection Based on Neural Network Optimized by Genetic Algorithm" en: 8–11.

9. Logeswari, G, S Bose, and T Anitha. 2023. "An Intrusion Detection System for SDN Using Machine Learning." https://doi.org/10.32604/iasc.2023.026769.

10. Makarand, L. 2022. "Machine Learning Applications in Engineering Education and Management Intrusion Detection System Attack Detection and Classification Model with Feed-Forward LSTM Gate in Conventional Dataset" 02 (01): 20–29.

11. Mirsky, Yisroel, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. "Kitsune : An Ensemble of Autoencoders for Online Network Intrusion Detection," no. February: 18–21.

12. Pranto, Badiuzzaman, Hasibul Alam Ratul, Mahidur Rahman, and Ishrat Jahan Diya. 2022. "Performance of Machine Learning Techniques in Anomaly Detection with Basic Feature Selection Strategy - A Network Intrusion Detection System" 13 (1). https://doi.org/10.12720/jait.13.1.36-44.

13. Ramasamy, Mathiyalagan, and Pamela Vinitha Eric. 2023. "A Tree Growth Based Forward Feature Selection Algorithm for Intrusion Detection System on Convolutional Neural Network" 12 (1): 472–82. https://doi.org/10.11591/eei.v12i1.4015.

14. Sabir, Maha, Jawad Ahmad, and Daniyal Alghazzawi. 2023. "A Lightweight Deep Autoencoder Scheme for Cyberattack Detection in the Internet of Things." https://doi.org/10.32604/csse.2023.034277.

15. Schmidt, Mark. 2020. "TheRepository at St . Cloud State Autoencoder-Based Representation Learning to Predict Anomalies in Computer Networks."

16. Shahid, Mustafizur R., Gregory Blanc, Zonghua Zhang, and Herve Debar. 2019. "Anomalous Communications Detection in IoT Networks Using Sparse Autoencoders." *2019 IEEE 18th International Symposium on Network Computing and Applications, NCA 2019*, 1–5.

https://doi.org/10.1109/NCA.2019.8935007.

17. Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. 2018. "A Deep Learning Approach to Network Intrusion Detection." *IEEE Transactions on Emerging Topics in Computational Intelligence* 2 (1): 41–50. https://doi.org/10.1109/TETCI.2017.2772792.

18. Shukla, Rakhi, and Aarti Kumar. 2023. "Security Enhancement Model for Intrusion Detection System Using Classification Techniques :" 5 (1): 125–32. https://doi.org/10.35629/5252-0501125132.

19. Siddique, Kamran, Zahid Akhtar, Farrukh Aslam Khan, and Yangwoo Kim. 2019. "KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research." *Computer* 52 (2): 41–51. https://doi.org/10.1109/MC.2018.2888764.

20. Sindian, Samar, and Samer Sindian. 2020. "An Enhanced Deep Autoencoder-Based Approach for DDoS Attack Detection 3 Deep Neural Network 2 Related Work" 15: 716–24. https://doi.org/10.37394/23203.2020.15.72.

21. Song, Tianbao, Jingbo Sun, B O Chen, Weiming Peng, and Jihua Song. 2019. "Latent Space Expanded Variational Autoencoder for Sentence Generation." *IEEE Access* 7: 144618–27. https://doi.org/10.1109/ACCESS.2019.2944630.

22. Song, Youngrok, Sangwon Hyun, and Yun Gyung Cheong. 2021. "Analysis of Autoencoders for Network Intrusion Detection†." *Sensors* 21 (13): 1–23. https://doi.org/10.3390/s21134294.

23. Wang, Chao, Hongri Liu, Yunxiao Sun, Yuliang Wei, Kai Wang, and Bailing Wang. 2022. "Dimension Reduction Technique Based on Supervised Autoencoder for Intrusion Detection of Industrial Control Systems" 2022.

24. Xu, W E N, Julian Jang-jaccard, Amardeep Singh, and Fariza Sabrina. 2021. "Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset." *IEEE Access* 9: 140136–46. https://doi.org/10.1109/ACCESS.2021.3116612.

25. Yu, Yang, Jun Long, and Zhiping Cai. 2017. "Network Intrusion Detection through Stacking Dilated Convolutional Autoencoders." *Security and Communication Networks* 2017. https://doi.org/10.1155/2017/4184196.

26. Zhang, Baoan, Yanhua Yu, and Jie Li. 2018. "Network Intrusion Detection Based on Stacked Sparse Autoencoder and Binary Tree Ensemble Method." *2018 IEEE International Conference on Communications Workshops, ICC Workshops 2018 - Proceedings*, no. 61702046: 1–6. https://doi.org/10.1109/ICCW.2018.8403759.