# Ensembling of Attention-based Recurrent Units for Detection and Mitigation of Multiple Attacks in Cloud

Kalaivani M[1], Padmavathi G[2]
Ph.D. Scholar, Department of Computer Science,
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, India.[1]
Dean-School of Physical Sciences and Computational Sciences and Professor in the Department of Computer Science,
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, India[2]

***Abstract*—In the recent years, number of threats to network security increases exponentially as the Internet users which poses serious threat in cloud storage application. Detection and defending against the multiple threats are currently a hot topic in industry and considered as one of the challenging research in academia. Many methodologies and algorithms devised to predict the different attacks. Still, most of the methods cannot simultaneously achieve high performance of prediction with a small number of false alarm rates. In this scenario, Deep Learning (DL) algorithms are appropriate and intelligent to categorize the multiple attacks. Still, most of the existing DL techniques are computationally inefficient that may degrade the performance in predicting the both normal and attack information. To overcome this aforementioned problem, this paper proposes the hybrid combination of attention maps with deep recurrent networks to mitigate the multiple attacks with low computational overhead. Initially, the pre-processing step is proposed to the inputs in a specified range. Later on, input data are fed into the Attention Enabled Gated Recurrent Networks (AEGRN) which is used to remove the redundant features and select the optimal features that aids for the better classification. Further to enhance the faster response, deep feed forward layers are proposed to replace the traditional deep neural networks. Numerous metrics for performance, including accuracy, precision, recall, specificity, and F1-score, are examined and analyzed as part of the thorough experimentation utilizing multiple datasets, including NSL-KDD-99, UNSW -2019, and CIDC-001. Comparisons of performance between the method that is suggested and existing models developed with DL are used to demonstrate the proposed algorithm's supremacy. The suggested framework surpasses the other DL models and has the best accuracy in predicting with little computational overhead, according to an investigation.**

***Keywords*—*Multiple threats; deep learning algorithm; attention enabled gated recurrent networks; NSL-KDD; UNSW; CIDC-001***

## I. Introduction

Internet Based Communication is used for managing big industry and transformed the scenario of monitoring and interaction methodology. Its scope of services also included the medical industry and was applicable to banking, schooling, government departments, military, and recreation. In addition, time, network development gives hackers and intruders opportunities to find illegal ways to break into an organization. Multiple assaults that have the capacity to deny services to legitimate customers are one of the main risks to the IP network on which numerous researchers have focused their attention. Therefore, maintaining the security and protection of various websites operating on the Internet is primarily required of the secured network [1-2]. Due to their qualities, such as quick access and primitive ways of attack detection, these attacks have expanded significantly.

It can be difficult to distinguish between malicious and lawful network data because intruders have unexpected behavior [3-5]. Applications run through anytime, anything, and anywhere in an internet context and interact remotely with a variety of devices or appliances. This makes it easier for bad actors to get devices. Despite these guidelines, interruption of devices or assistance is likely to be the first stage of many attacks due to factors such as ease of comprehension, simplicity of execution, lack of extensive technical knowledge on the part of the attacker, and variety of platforms and applications for aided attack orchestration [6–8].

These attacks can be single-source assaults commencing at just one host or multi-source attacks that distribute attack packages to the target across numerous hosts. Also, attack toolkits have been developed and therefore are easily accessible in online today [9–10]. But these tools can be exploited by the intruders to enforce the attacks with least effort. As a result, more examinations are performed in recent years through the use of numerous algorithms to develop the defensive system for cloud attacks. But these traditional systems possess the various problems such as high memory, high bandwidth and processing capacity. It is vital to design Intrusion Detection Systems in order to counteract this lack of security assaults, in which the network attacks can be prevented primarily. With exponential increase, information is steadily moved from separate networks. IDS needs improvisation in predicting the intrusion in such huge data environment.

IDS has been using deep computing and machine learning techniques in the last ten years to help classify the observed data using known characteristics or attributes that have been learned from training datasets. The purpose of ML and DL-based techniques, which have some limitations, is to evaluate network traffic packet properties and set a reasonable threshold for separating attacks from genuine traffic. For instance, tatistical recognition methods [15], Neural Networks [11], Support Vector Machine [12], nearest neighbor [13] andclustering [14]. These current studies reveal that various studies have been conducted to offer treatments to deal with this difficulty by outlining particular treatments for emerging network assaults.

Since these intelligent-based IDS have only recently been introduced, a number of issues need to be resolved. Here are a few of the current issues that studies on attack detection systems are now facing:

*1)* The majority of currently used techniques concentrate on identifying a single attack with a low false alarm rate, although they typically fall short of reaching a high detection rate.

*2)* Knowing the many characteristics of attacks is important, but identifying the ones that can really help in the detection of assaults is even more crucial. However, because of redundant information and excessive computational expense, certain existing techniques commonly have high false positive rates. A well-organized network attack detection method remains a promising research subject because earlier methods also fall short in terms of attaining efficient accuracy.

Considering these problems, this research article proposes the novel integration of the Attention layers with the Gated recurrent NN to achieve the high classification ratio in mitigating the network attacks with less computational complexity. Following are the paper's main contributions:

*1)* Self-Attention Maps are introduced in Gated Recurrent Neural Network (RNN) to achieve the better feature selection that in returns support for the better detection ratio.

*2)* Data-Pre-processing technique is employed for the increasing the speed in detecting the attacks.

*3)* Feedforward Learning Layers- They are introduced in the place of the conventional neural networks to achieve the faster training with less error detection.

Following is how the manuscript is organised: Details on the background and related works are provided in Section II. The description of the dataset, data pre-processing, and suggested approach are shown in Section III. The following Section IV provides further details on the experimental findings. The Section V provides a conclusion and future enhancements.

## II. History and Related Works

Abirami et al. (2022) demonstrated how “Deep Reinforcement Learning (DRL)” might be used in a cloud network to offload tasks while also recognizing generalized attackers. Techniques for identity-based linear classification are used in virtual machine attack categorization channels. This proposed system supports methods for remote information analysis. Reinforcement learning has the potential to reduce data secrecy and improve cooperation. The sole drawback of this system is the prolonged computation time [16].

In 2022, Tao et al., developed a “Continuous Duelling Deep Q-Learning (C-DDQN)” technique for protecting the cloud. The suggested Dynamic Field Adaptive System and improving are the fundamental ideas of this system. The convergence and learning capabilities of the aforementioned structure are preferable than those when transfer learning methods weren't used. But this framework's primary problem is the rising energy consumption [17].

Recurrent and convolution neural networks were combined in 2021 by Hizal et al. to create a DL method for threat detection in security of the cloud. Any discovered or forbidden traffic cannot be sent to the cloud server using this method. The recommended method is 99.86% accurate for classification into five classes. But this framework's primary drawback is the higher connectivity cost [18].

In 2020, Karri et al. proposed a three-stage abnormality detection framework that utilized DL for intrusion attack detection. CNN, GANomaly, and K-means clustering algorithms are all used by the system. The effectiveness of the network and automated intrusion detection had been either greatly improved. The main advantage of the aforementioned structure is that it reduces the level of computation without reducing cost [19].

By Wang et al. in 2022, stacking contractive auto encoder (SCAE) system was unveiled. The Support Vector Machine configuration serves as the framework's core. By using the unfiltered network information, this structure enables the automatic learning of improved as well as more trustworthy low-dimensional properties. This paradigm significantly reduces the analytical complexity. This technique leads to improved detection efficiency. This framework's drawback, nevertheless, is that it cannot be used in contexts where events happen in the present [20].

PredictDeep was introduced in 2020 as an approach for prediction of anomaly in big data environments by Elsayed et al. GCNs, or Graph Convolutional Networks, form the basis of the system. This solution produced better outcomes in regards of the fast discovery and forecasting of incidents of security and was able to cope with the multifaceted nature of clouds. The problem with this technique, though, is that it doesn't recognize and classify irregularities in a range of classifications in accordance with the changes in system function they cause [21].

Nguyen et al. (2021) examined the difficulties associated with compute offloading and cybersecurity in a multiple-user-friendly mobile edge-cloud computing framework utilizing blockchain. The above structure provides an effective authorization mechanism powered by blockchain that may protect servers in the cloud from incorrect offloading practices in order to boost offloading security. A complex DRL method called a double-dueling Q-network was developed by this framework to do this. This framework is lowering the latency, energy consumption, and intelligent contract fees. But this approach has the drawback that efficiency degrades as the amount of information increases [22].

RNN-based DL approaches were examined by Kimmel et al. (2021) for their efficacy in identifying malware in cloud.

The focus of the framework was on LSTMs and bidirectional RNNs. Such frameworks progressively understand malware behaviors based on the course of operation, minute activities, and system statistics such as CPU, memory, and disc usage. With this architecture, there are high detection rates. but, cannot maintain the identical degree of performance when dealing with diverse data [23].

Loukas et al. (2018) introduced a recurrent NN with a deep multilayer perceptron architecture which is capable of understanding the temporal context of several attacks. A computational framework was developed to determine whether compute offloading is favorable utilizing detection latency as the criterion, given networking operational parameters and DL framework processor needs. When the processing requirements are more severe and the network has become more reliable, offloading lowers detection delay to a greater extent. The biggest problem with this structure though, is the additional communication complexity [24].

By fusing a Convolutional NN with Grey Wolf Optimization, Garg et al. (2019) created a composite data mining method for identifying network abnormalities. The GWO and CNN learning procedures were improved in order to enhance the framework's abilities for initial sample creation, exploring, taking advantage of and discarding functionality. The above structure works better in terms of precision, false alarms, and recognition rate. This strategy does have a disadvantage, too, in that it increases computing difficulty [25]. Table I following provides an overview of several relevant studies.

TABLE II. LIST OF RELATED WORKS FROM LITERATURE

| Author's name | Proposed methodology | Merits | Demerits |
|---|---|---|---|
| Abirami et al., (2022) | DRL | Minimises the data secrecy | Increased computational delay |
| Tao et al. (2022) | Continuous duelling deep Q-learning | Fast convergence | Increased energy consumption |
| Hizal et al., (2021) | K-means clustering, GANomaly and CNN algorithms | Reduced the computational complexity | fails to lower down on time overhead |
| Karri er al., | GANomaly and CNN algorithms | Reduced the computational complexity | However, fails to reduce time overhead |
| Wand et al., (2022) | Stacked Contractive Autoencoder and Support Vector Machines (SVM) | Reduced the analytical overhead | Not suitable for real time environment |
| Elsayed et al., (2022) | Graph Convolution Networks (GCNs) | Timely detection and prediction of security breaches | It does not predict and classify anomalies under change in the system behaviour |
| Nguyen et al.,(2021) | Mobile edge-cloud computation offloading system | Minimized the long-term system costs of latency, energy consumption | Performance gets degraded when the data is increased |
| Kimmel et al., | LSTM and Bidirectional RNNs (BIDIs) | Achieves high detection rates | Does not handle heterogeneous data |
| Loukas et al., (2018) | MLP and RNN | Reduction in detection latency | Increased communication overhead |
| Gard et al., (2019) | Grey wolf Optimization (GWO) and (CNN) | High accuracy and high detection rate | Increased computational complexity |

## III. PROPOSED ARCHITECTURE

According to Fig. 1, the hybrid suggested network's framework is made up of three sub modules. In the first module, multiple datasets are pre-processed and inputted to the proposed network. The second module consists of the proposed SA-GRU-FF framework in which attention layer is integrated to remove redundant and non-optimal temporal features. These features are then fed into the fully connected deep feed forward networks based on Extreme Learning Machines (ELM) for classification of the multiple attacks.

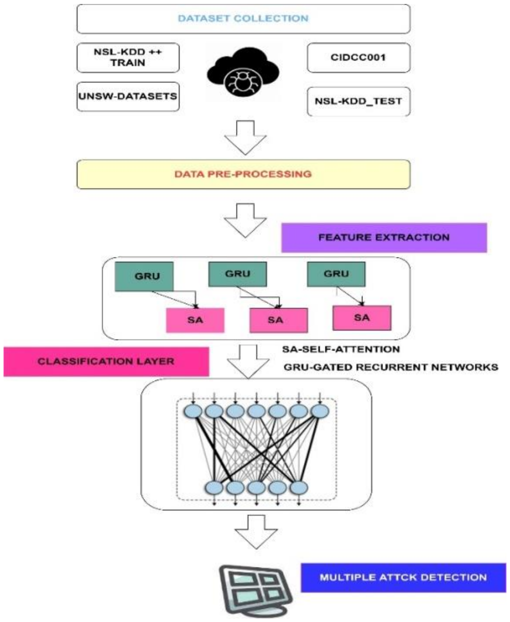


Fig. 1. Proposed architecture for the GRU-SA-FF based multiple classifications of attacks.

### A. Materials and Methods

Three distinct datasets, namely CIDDS-001 [27], UNSW-NB15 [28], and NSLKDD [29], are employed in this investigation. We choose the CIDDS-001 and UNSW-NB15 datasets because they are the most current statistics produced and include real data traffic, which makes them beneficial for designing accurate IDSs for tracking and finding novel forms of denial of service attacks in cloud networks. An IDS based on anomalies may now be created with the help of the CIDDS-001 dataset, which was just made accessible. In all, the collection contains around 32 million tracks, covering both normal and attack traffic. This dataset is composed of 12 identifying features and two distinguishing traits. Random sampling is applied to acquire 80,000 normal and 20,000 DoS attack events from the relational database of server traffic data, totaling 100,000 events. Using the extracted sample, the cross-fold validity and hold-out of the classifiers are tested. A new contribution to the public domain, the UNSW-NB15 dataset, was also utilized for the purposes of testing. In the dataset, there are 49 characteristics and 1 class attribute. A subset of the dataset uses the training and test establishes, "UNSW NB15 Train & UNSW NB15 Test". There are 175,341 occurrences in the train set compared to 82,332 in the test set. "There are 56,000 occurrences of ordinary traffic and 119,341 illustrations of attack traffic on the platform set. Additionally, there are 37,000 examples of ordinary traffic and 45,332 cases of attack traffic in the test set". Hold-out confirmation makes use of both the whole train set and the test set, while cross-fold assessment solely utilizes the set that has been tested. The NSL-KDD dataset is then utilized to do classifier validation as well. 41 measures including 1 class attribute are part of the dataset. The NSL KDD dataset's KDDTrain+ (training) as well as KDDTest+ (testing) sets are utilised in this study. 13,499 attack traffic instances and 11,743 regular traffic instances make up the total 25,192 instances in the KDDTrain+ set. While the KDDTest+ set has a total of 22,544 instances, including 12,833 instances of regular traffic and 9,711 incidents of attack traffic. On each dataset separately, hold out as well as cross fold validation of classifers are performed. The selection of these sets was made to prevent randomly selecting cases from the entire NSL-KDD dataset.

### B. *Data Reorganizing*

The input data are first analysed, and then they are fed into a standardization approach, which assists to convert the bulk of attributes with numerical data to a specified numeric domain. Min-Max normalisation is used in conjunction with the linear transformation concept to accomplish this. After pre-processing step, new pre-processed datasets is formed from the original raw datasets. These pre-processed data is given for feature extraction module.

### C. *Feature Extraction using Self –Attention Gated Recurrent Networks*

The operation of gated recurrent sections, self-attention, and mixed combinations of self-attention gated recurrent units are covered in this section.

*1) Gated recurrent units – An overview*: One of most interesting form of LSTM is known as GRU the architecture is depicted in Fig. 2. The forget gate with input vector are intended to be combined into a single vector according to the concept set out by Chung et al. [30]. Both long-term sequences and memories are supported by this network. When contrasted to the LSTM network, the complexity is drastically reduced.

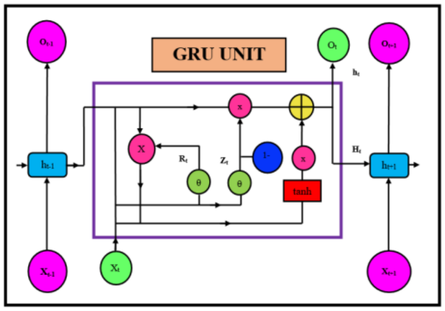


Fig. 2. GRU's architecture.

Chung developed the following equations to illustrate the traits of GRU.

$$h_t = (1 - x_t) \odot h_{t-1} + x_t \odot h_t \quad (1)$$

$$\widetilde{h_t} = g(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h \quad (2)$$

$$z_t = \sigma(W_h x_t + U_z h_{t-1} + b_z) \quad (3)$$

$$r_t = \sigma(W_h x_t + U_r h_{t-1} + b_r) \quad (4)$$

The following is the general GRU characteristic equation:

$$R = GRU(\sum_{t=1}^{n}[x_{t,} h_{t,} z_{t,} r_t\big(W(t), B(t), \eta(tannh)\big)] \quad (5)$$

where, "xt ⇨input feature at the present state, yt ⇨output state , ht ⇨ output of the unit as of this moment, Zt & rt⇨update & reset gates, W(t) ⇨ weights, B(t) ⇨ bias weights at present instant".

*2) Self-awareness maps:* In 2014, the attentive map was proposed to describe the appropriate words in a sequence-to-sequence structure. In the vast mainstream of contemporary works, redundant characteristics that support accurate categorization mechanisms are imitated using attention layers. The self-attention process, commonly alluded to as the intra-attention procedure, generates the three vectors Q, K, and V for each input pattern. Thus, the results sequences are created by transforming the input patterns from all of the layers. It is a technique that, in its simplest form, maps the query string to the set of key-pair collections using logarithmic dot processes. The mathematical formula that follows can be used to get the dot multiplying for self-attention.

$$F(K, Q) = ((K), Q^T))/(V_K)^\wedge 0.5 \quad (6)$$

### D. *Proposed Feature Extraction*

BiGRU networks, which combine forward and backward GRU, are built for gathering meaningful information from the many dataset streams. Eq. (9) delivers data on the precise properties of the BiGRU network. In order to classify data, the BiGRU network collects spatiotemporal characteristics that incorporate a variety of different pieces of data. Although the training time, which makes up the overhead in the classification layer, may be affected by the more varied information in these characteristics. Self-attention layers that are inserted among the BiGRU network and classification layer help to diminish the resulting classification cost. Eq. (6) is used to create the attention characteristics retrieved from the input features of the BiGRU network that are then given to the feed-forward level of classification via the softmax layer.

$$P(F) = GRU\big(\sum_{t=1}^{n}[x_{t,} h_{t,} z_{t,} r_t\big(W(t), B(t), \eta(tannh)\big)\big] \quad (7)$$

$$P(B) = GRU\big(\sum_{t=1}^{n}[x_{t,} h_{t,} z_{t,} r_t\big(W(t), B(t), \eta(tannh)\big)\big] \quad (8)$$

Combining the Eq. (7) and Eq. (8)

$$P(BiGRU) = P(F) + P(B) \quad (9)$$

The following information is related to integrated Self-Attention (SA) with BiGRU feature extraction.

$$Y = Softmax\ \big(P\big(BiGRU\big), F(K, Q)\big) \quad (10)$$

### E. Feed Forward Classification Layers

After receiving these attributes for the fully connected forward feed-forward network, the final classifying is carried out. Layers are entirely linked using the ELM principle. The principle of auto-tuning capacity underlies the operation of a particular class of neural network known as an ELM, which only uses one hidden unit. In regards to dependability, speed, and computational burden, ELM performed better than other learning models like “Support vector machines (SVM), Bayesian Classifier (BC), K-Nearest Neighbourhood (KNN), and even Random Forest”.

There is just one hidden layer in this specific neural network; therefore it may not require to be modified. Compared to other learning algorithms like Random Forest and Support Vector Machines, ELM operates better, more quickly, and with lower computational cost. Small training error and improved approximation are the ELM's main benefits. ELM uses non-zero activation functions and weight biases that are automatically tuned. The ELM's intricate operating mechanism is covered in [26]. Following Attention maps, the ELM's input features maps are represented by:

$$X = F(Y) \tag{11}$$

where, Y ⇨ features from Self Attention BiGRU network ,

*The ELM’s output function is represented by the symbol*

$$Y(n) = \ X(n)\beta = X(n)X^T(\frac{1}{C}XX^T)^{-1}O \tag{12}$$

ELM's comprehensive training is provided by:

$$S = \alpha(\sum_{n=1}^{N}(Y(n), B(n), W(n)) \tag{13}$$

Finally, the softmax activation layers are applied for the above feedforward layers to achieve the best accuracy.

## IV. Experimentation Details

The entire algorithm was designed on an Intel Workspace with a 3.2 GHz of frequency, I7 CPU (NVIDIA GPU) and a16GB of RAM. Utilizing Keras (Tensorflow) as the rear end, the suggested baseline infrastructure was created.

### A. Performance Metrics

Deep feed forward training networks that classify the necessary classifications into typical sensitive and malicious information as well as the suggested design are validated as part of the experiment. Metrics including “accuracy, sensitivity, selectivity, recall, and f1-score” are used to gauge the suggested design's effectiveness. The calculations for the metrics used to assess the suggested architecture are shown in Table II in their respective computation formulae. Additionally, Table III shows the experimental hyperparameters that were utilized to train the suggested network.

### B. Results and Discussion

The experimentation is carried out based on component structures with the same parameters as the proposed framework. In detail, the existing structures were one dimensional Long Short Term Memory [30], Gated Recurrent Units [31], Optimized LSTM [23], and BiGRU [32]. The technique was validated and a comparison study was performed using four different datasets.

TABLE III. ALGEBRAIC EQUATIONS FOR THE CALCULATION OF PERFORMANCE METRICS

| SL. NO | Validation Metrics | Formulae |
|---|---|---|
| 01 | Accuracy ($A_{cc}$) | $\frac{TP + TN}{TP + TN + FP + FN}$ |
| 02 | Sensitivity or recall ($R_{ll}$) | $\frac{TP}{TP+FN}$x100 |
| 03 | Specificity ($S_{ty}$) | $\frac{TN}{TN + FP}$ |
| 04 | Precision ($P_{cn}$) | $\frac{TN}{TP + FP}$ |
| 05 | F1-Score ($F_{cr}$) | $\frac{Precision * Recall1}{Precision + Recall1}$ |

Where, “TP – True Positive Values, TN – True Negative Values, FP – False Positive and FN – False Negative”

TABLE IV. HYPER PARAMETERS USED IN THE NETWORK'S TRAINING

| SL. NO | Hyper-Parameters | Specifications |
|---|---|---|
| 1 | GRU cell count | 10 |
| 2 | Epochs count | 200 |
| 3 | Batch Size | 30 |
| 4 | Learning Rate | 0.001 |
| 5 | Momentum | 0.2 |
| 6 | Dropouts | 0.2 |

TABLE V. USING THE CIDCC-001 DATASETS, EFFICIENCY STATISTICS OF THE DISTINCT ALGORITHMS

| Algorithms | Validation Metrics | | | | |
|---|---|---|---|---|---|
| | $A_{cc}$ | $P_{cn}$ | $R_{ll}$ | $S_{ty}$ | $F_{cr}$ |
| LSTM | 0.89 | 0.85 | 0.834 | 0.190 | 0.84 |
| GRU | 0.91 | 0.86 | 0.856 | 0.1556 | 0.857 |
| Optimized-GRU | 0.92 | 0.89 | 0.887 | 0.1290 | 0.885 |
| Proposed Model | 0.98 | 0.97 | 0.966 | 0.11 | 0.975 |

TABLE VI. USING UNSW2019 DATASETS, MONITORING OF THE MULTIPLE ALGORITHMS

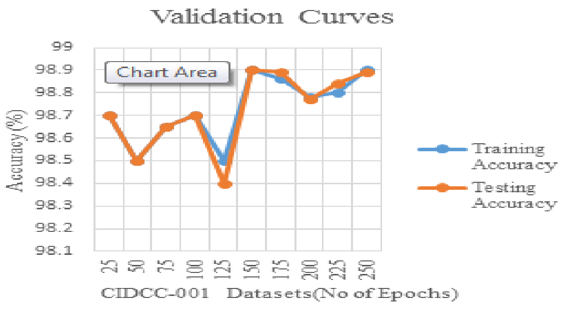
| Algorithms | Validation Metrics | | | | |
|---|---|---|---|---|---|
| | $A_{cc}$ | $P_{cn}$ | $R_{ll}$ | $S_{ty}$ | $F_{cr}$ |
| LSTM | 0.874 | 0.87 | 0.864 | 0.150 | 0.86 |
| GRU | 0.902 | 0.90 | 0.89 | 0.110 | 0.91 |
| Optimized-GRU | 0.910 | 0.91 | 0.90 | 0.100 | 0.92 |
| Proposed Model | 0.983 | 0.98 | 0.974 | 0.011 | 0.983 |

TABLE VII. NSL-KDD+(TRAIN) DATASETS PERFORMANCE INDICATORS OF THE SEVERAL ALGORITHMS

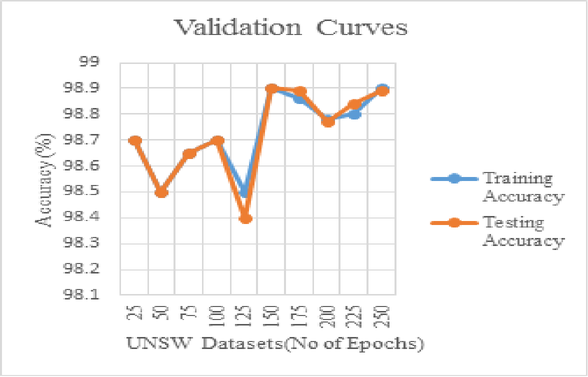
| Algorithms | Validation Metrics | | | | |
|---|---|---|---|---|---|
| | $A_{cc}$ | $P_{cn}$ | $R_{ll}$ | $S_{ty}$ | $F_{cr}$ |
| LSTM | 0.88 | 0.875 | 0.834 | 0.190 | 0.84 |
| GRU | 0.92 | 0.90 | 0.856 | 0.1556 | 0.857 |
| Optimized-GRU | 0.93 | 0.92 | 0.887 | 0.1290 | 0.885 |
| Proposed Model | 0.988 | 0.98 | 0.974 | 0.001 | 0.980 |

TABLE VIII. NSL-KDD+(TEST) DATASETS PERFORMANCE METRICS FOR THE DIFFERENT ALGORITHMS

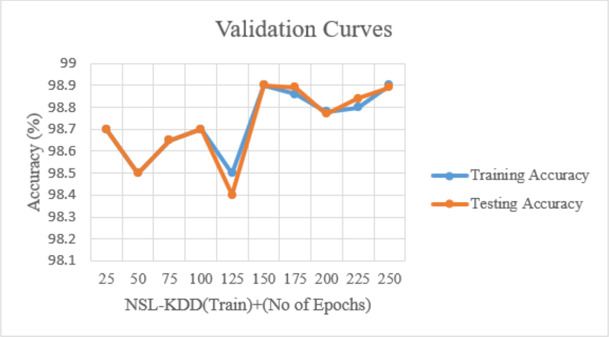
| Algorithms | Validation Metrics | | | | |
|---|---|---|---|---|---|
| | $A_{cc}$ | $P_{cn}$ | $R_{ll}$ | $S_{ty}$ | $F_{cr}$ |
| LSTM | 0.88 | 0.875 | 0.834 | 0.190 | 0.84 |
| GRU | 0.92 | 0.90 | 0.856 | 0.1556 | 0.857 |
| Optimized-GRU | 0.93 | 0.92 | 0.887 | 0.1290 | 0.885 |
| Proposed Model | 0.988 | 0.98 | 0.974 | 0.001 | 0.980 |



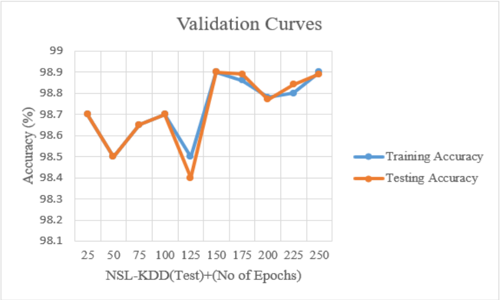
(a)



(b)



(c)



(d)

Fig. 3. Validation performance of the suggested model using distinctive datasets a) CIDCC-001 datasets b) UNSW-datasets c) NSL-KDD datasets (Train) d) NSL-KDD datasets (Test).

Tables IV, V, VI, and VII demonstrates the proposed algorithm's performance metrics for categorizing several assaults using various datasets. The Table IV represents, the outcomes of proposed and existing frameworks when testing under CIDCC-001 Datasets. The Table V, Table VI and Table VI represents, the outcomes of proposed and existing model when testing under UNSW2019, NSL-KDD+(Train) and NSL-KDD+(TEST) datasets respectively. From Table IV, V, VI and VII, it is observed that, the suggested model GRU-SA-FF has demonstrated the best performance in detecting the numerous attacks. The integration of Self-attention maps has provided the best results in contrast to different DL techniques. Additionally, the validation effectiveness of the suggested model (see Fig. 3) is assessed using various datasets, and it is discovered that the RMSE (root mean square error) in between training and testing data is 0.001.

TABLE IX. MBT IN SUPPORT OF DIFFERENT ALGORITHMS USING DIFFERENT DATASETS

| Datasets | (MBT)-secs | | | |
|---|---|---|---|---|
| | LSTM | GRU | Op-LSTM | Proposed Model |
| CIDCC001 | 0.5 | 0.45 | 0.37 | 0.23 |
| UNSW | 0.45 | 0.39 | 0.31 | 0.21 |
| NSL-KDD++ Train | 0.5 | 0.45 | 0.37 | 0.22 |
| NSL-KDD++ Test | 0.43 | 0.42 | 0.38 | 0.22 |

Model building times for various classifiers are shown in Table VIII for four datasets employing hold-out evaluation. Recognising how essential it is to deliberate how long a system needs train until it is successful at spotting various risks, the main driver aimed at estimating MBT is this realisation. Because of this, MBT helps to achieve a good trade-off among computational complexity and the accuracy of classifiers. The suggested model's average MBT when trained on the different sets of data is 0.22s, according to the above table, compared to 0.36s, 0.41s, and 0.48s for Op-LSTM, GRU, and LSTM, respectively, for Op-LSTM. According to the evaluation, the suggested framework uses only 0.22 seconds and excels at designing countermeasures against several threats.

## V. CONCLUSION AND FUTURE ENHANCEMENT

In this work, investigation on integration of Self-attention maps with GRU for securing the cloud against the multiple attacks is carried out. The role of self –attention network with the BiGRU to select the optimal features that can aid for the classification layers is proposed in this paper. Additionally, role of feed forward layers which works on principle of ELM has been used in the proposed research to achieve the better classification with reduced computational burden and quick speed. Precision, specificity, susceptibility, false alarm rate, and region under the curve of receiver operating characteristics are used to assess the performance of the suggested model. On the CIDDS-001, UNSW-NB15, & NSL-KDD datasets, all of the classifiers are benchmarked. Results demonstrate in terms of a superior detection ratio and so little overhead, the proposed approach have done better over the other DL models. As the future scope, performance of the proposed model is required for the validation with real time datasets and also brighter light of deploying in the resource constraint in Cloud.

## REFERENCES

[1] Laghrissi, F., Douzi, S., Douzi, K. et al., “IDS-attention: an efficient algorithm for intrusion detection systems using attention mechanism.” J Big Data, Vol 8, 149, 2021.

[2] Maha M, Althobaiti K, Mohan KP, Deepak G, Sachin K, Mansour RF. “An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems.” Measurement. 2021, 186(110145):0263–2241.

[3] Anthi E, Javed A, Rana O, Theodorakopoulos G “Secure data sharing and analysis in cloud-based energy management systems.” In Cloud Infrastructures, Services, and IoT Systems for Smart Cities, pages 228–242. Springer, 2017

[4] Baykara, M., & Das, R. “A novel hybrid approach for detection of webbased attacks in intrusion detection systems.” International Journal of Computer Networks and Applications, 4(2), 62–76, 2017

[5] Bergstra, J., &Bengio, Y. (2012). “Random search for hyper-parameter optimization.” Journal of Machine Learning Research, 13(Feb), 281–305.

[6] Mahboob AS, Moghaddam MRO. “An Anomaly-based Intrusion Detection System Using Butterfy Optimization Algorithm,” 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), 2020; pp. 1-6

[7] BButun, I., Morgera, S. D., &Sankar, R.. “A survey of intrusion detection systems in wireless sensor networks.” IEEE Communications Surveys & Tutorials, 16(1), 266–282, 2014

[8] Chen, T., &Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In ACM, proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 785–794).

[9] Khan MA. HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system. Processes. 2021; 9(5): 834.

[10] Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y. An ensemble method based on selection using bat algorithm for intrusion detection. Comput J. 2018;61(4):526–38.

[11] Demšar, J. (2016). Statistical comparisons of classifers over multiple data sets. Journal of Machine Learning Research, 7(Jan), 1–30.

[12] Dhanjani, N. (2013). Hacking lightbulbs: Security evaluation of the philips hue personal wireless lighting system. Retrieved November 3, 2019, from https://www.dhanjani.com/docs/Hacking

[13] Diro, A. A., &Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 82, 761–768.

[14] Girma A, Garuba M, Goel R. Advanced machine language approach to detect DDoS attack using DBSCAN clustering technology with entropy. In: Latifi S, ed. Information Technology - New Generations. Advances in Intelligent Systems and Computing, 2018, vol. 558. Cham, Switzerland: Springer, pp. 125–131

[15] Douglas, P. K., Harris, S., Yuille, A., & Cohen, M. S. (2011). Performance comparison of machine learning algorithms and number of independent components used in FMRI decoding of belief vs. disbelief. Neuroimage, 56(2), 544–553.

[16] P. Abirami, S. Vijay Bhanu and T. K. Thivakaran, "Crypto-Deep Reinforcement Learning Based Cloud Security for Trusted Communication," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, pp. 1-10, doi: 10.1109/ICSSIT53264.2022.9716429.

[17] Y. Tao, J. Qiu and S. Lai, "A Hybrid Cloud and Edge Control Strategy for Demand Responses Using Deep Reinforcement Learning and Transfer Learning," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 56-71, 1 Jan.-March 2022, doi: 10.1109/TCC.2021.3117580.

[18] S. Hizal, Ü. ÇAVUŞOĞLU and D. AKGÜN, "A new Deep Learning Based Intrusion Detection System for Cloud Security," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021, pp. 1-4, doi: 10.1109/HORA52670.2021.9461285.

[19] C. Karri and M. S. R. Naidu, "Deep Learning Algorithms for Secure Robot Face Recognition in Cloud Environments," 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2020, pp. 1021-1028, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00154.

[20] W. Wang, X. Du, D. Shan, R. Qin and N. Wang, "Cloud Intrusion Detection Method Based on Stacked Contractive Auto-Encoder and Support Vector Machine," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1634-1646, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.3001017.

[21] M. A. Elsayed and M. Zulkernine, "PredictDeep: Security Analytics as a Service for Anomaly Detection and Prediction," in IEEE Access, vol. 8, pp. 45184-45197, 2020, doi: 10.1109/CCESS.2020.2977325.

[22] D. C. Nguyen, P. N. Pathirana, M. Ding and A. Seneviratne, "Secure Computation Offloading in Blockchain Based IoT Networks With Deep Reinforcement Learning," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 3192-3208, 1 Oct.-Dec. 2021, doi: 10.1109/TNSE.2021.3106956.

[23] J. C. Kimmel, A. D. Mcdole, M. Abdelsalam, M. Gupta and R. Sandhu, "Recurrent Neural Networks Based Online Behavioural Malware Detection Techniques for Cloud Infrastructure," in IEEE Access, vol. 9, pp. 68066-68080, 2021, doi: 10.1109/ACCESS.2021.3077498.

[24] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon and D. Gan, "Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning," in IEEE Access, vol. 6, pp. 3491-3508, 2018, doi: 10.1109/ACCESS.2017.2782159.

[25] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya and R. Ranjan, "A Hybrid Deep Learning-Based Model for Anomaly Detection in Cloud Datacenter Networks," in IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 924-935, Sept. 2019, doi: 10.1109/TNSM.2019.2927886.

[26] Verma, A., &Ranga, V. (2018). Statistical analysis of CIDDS-001 dataset for network intrusion detection systems using distance-based machine learning. Procedia Computer Science, 125, 709–716.

[27] Verma, A., &Ranga, V. (2019a). ELNIDS: Ensemble learning based network intrusion detection system for RPL based Internet of Things. In 2019 4th International conference on Internet of Things: Smart innovation and usages (IoT-SIU) (pp. 1–6). IEEE.

[28] Verma, A., &Ranga, V. (2019). Evaluation of network intrusion detection systems for RPL based 6LoWPAN networks in IoT. Wireless Personal Communications, 108(3), 1571–1594.

[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv preprint arXiv:1412.3555, 2014

[30] Staudemeyer RC. 1 Applying long short-term memory recurrent neural networks to intrusion detection. South AfrComput J. 2015;56(1):136–54.

[31] Kim J, Kim J, Thu HLT, and Kim H. Long short term memory recurrent neural network classifer for intrusion detection, In Proc. Int. Conf. Platform Technol. Service (PlatCon); 2016, pp. 1–5.

[32] Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y. An ensemble method based on selection using bat algorithm for intrusion detection. Comput J. 2018;61(4):526