Musik in bayern

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

Classification and Prediction of Significant Cyber Incidents Using Optimized Deep Feed Forward logarithmic neural network

Mrs. Ajitha.I[1], Dr. A. Devi[2]
[1] Ph.D Research Scholar , Department of Computer Science, Dr.SNS Rajalakshmi College of Arts and Science, India
[2] Assistant Professor, Department of Computer Applications, Dr.SNS Rajalakshmi College of Arts and Science, India

To Cite this Article

Mrs. Ajitha.I , Dr. A. Devi " Classification and Prediction of Significant Cyber Incidents Using Optimized Deep Feed Forward logarithmic neural network"*Musik In Bayern, Vol. 88, Issue 10, Nov 2023, pp225-255*

Article Info

Received: 24-10-2023 Revised: 04-11-2023 Accepted: 14-11-2023 Published: 24-11-2023

Abstract: Given the shortcomings of deep learning (DL) and machine learning (ML) approaches for the prediction and classification of Significant Cyber Incidents (SCI), this work begins with the feature extraction stage of neural networks and constructs the Logarithm Neuron Layer (LNL) with more robust data processing capability. Since LNL operates logarithmic operations internally, unlike traditional neurons, it can analyze data without the need for data preprocessing phases. This reduces the influence of data normalization and other processes on the sample's original data and increases the neural network's accuracy. Additionally, the initial values of logarithmic operations might be improved based on the particular demands of the task. Utilizing LNL in conjunction with a deep neural network, the Optimized Deep feed forward logarithmic neural network (ODFFLOGNN) was created on that base, where Honey Bader Algorithm (HBA) for optimal adjustment of the hyperparameters. The significance of cyber security and the effect of COVID-19 on cyber security are clarified by this research. There are two subgroups of the dataset (referred to as SCI in the Center for Strategic and International Studies (CSIS) report): pre-pandemic SCI and post-pandemic SCI. Lastly, the research confirms that OFFLOGNN performs better than other methods for intrusion detection.

Keywords: Cyber Security, Significant Cyber Incidents, Neural Network, Logarithm Neuron Layer, Deep feed forward logarithmic neural network and Honey Bader Algorithm

1. Introduction

In recent years, the surge in both the quantity and complexity of cybersecurity attacks has propelled data mining (DM) methods into the spotlight for researchers. These techniques play a crucial role in detecting such attacks by analysing data and examining the side effects left by malware, spyware programs, and instances of network and host intrusions. Text analysis and mining, in particular, find widespread application in various cybersecurity domains, including malware detection and classification [1], malicious code detection ([2]), and predicting links ([3]). The rise of social media has further amplified the importance of text mining and examination, serving as essential tools for understanding users' contexts, such as their location, time, and integrating the information with characteristics of significant occasions, subjects, and passions.

Despite its popularity, the cybersecurity community has exhibited some reluctance in implementing an open method to security-related data, influenced by political factors and organizations' hesitancy to share data. Technical factors also play a role, with issues with quality, consistency, and an absence of consistency over the characteristics of factors to monitor or metrics for quantifying security data ([4]). Nevertheless, there is a noticeable shift in this trend with the emergence of large and open security datasets and data-sharing platforms supported by reputable organizations in the cybersecurity domain. Examples include VCDB ([5]), SecRepo

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

([6]), CAIDA ([7]), and others. This shift has led to an increased trend in using datasets to gain insights into how cyber incidents occur and how to defend against them. Significant Cyber Incidents (SCIs) are cyberattacks that have a major impact on individuals, organizations, or even entire nations. These attacks can cause widespread disruption, financial loss, and even physical harm. Here are some examples of recent SCIs:

- SolarWinds supply chain attack (2020): Hackers compromised the SolarWinds Orion network management software, which is used by many government agencies and private companies as shown in Fig.1 [8]. The attackersThis model re able to gain access to the networks of these organizations and steal data.

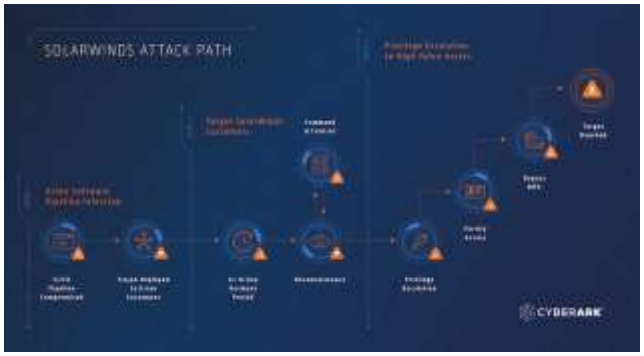

Fig.1. SolarWinds supply chain attack (2020)

- Microsoft Exchange Server vulnerabilities (2021): Hackers exploited vulnerabilities in Microsoft Exchange Server to gain access to email accounts and steal data as shown in Fig.2 [9]. The attacks affected tens of thousands of organizations around the world.

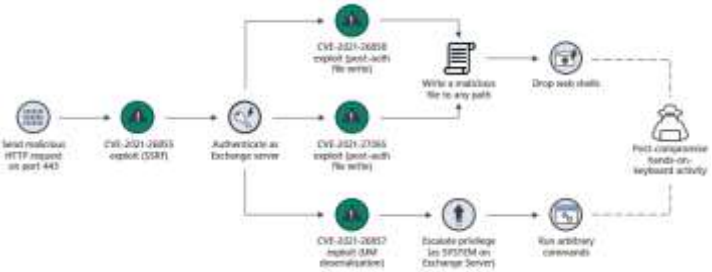

Fig.2. Microsoft Exchange Server vulnerabilities (2021)

- Cyberattacks against Ukraine (2022): Preceding and concurrent with the Russian invasion of Ukraine, a series of cyberattacks, likely orchestrated by Russian state-backed hackers, targeted Ukrainian infrastructure and government systems (Fig. 3 [10]).

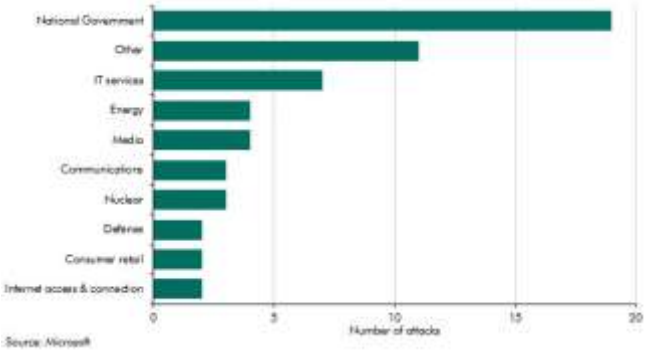

Fig.3. Cyberattacks against Ukraine (2022)

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

These are just a few examples of SCIs. The threat landscape is constantly evolving, and new vulnerabilities are discovered all the time. It is important for organizations and individuals to be aware of the latest threats and take steps to protect themselves from cyberattacks. The method of collecting knowledge and patterns from huge databases is called data mining [11]. In the context of SCI prediction, data mining can be used to analyze historical data about past SCIs in order to identify patterns and trends that can be used to predict future events. With this motivation, this work classifies SCI using ODFFLOGNN approaches. Here HBA is used for optimal adjustment of the hyperparameters. The centralized classifier is the research's purpose. The six continents of the planet are represented in the dataset. Instead of utilizing a different classifier for each continent, the dataset is employed to train the centralized classifier, leading to finally greater accuracy. This work determined, examined, and resolved the issue of predicting the continent's name utilizing the form of SCI. To accomplish the effectiveness of the classifiers for the gathered datasets is evaluated employing four different types of classifiers.

2. Related Work

The significance of cyber security and the effect of COVID-19 on cyber security were the main topics of this research [12]. Pre-pandemic SCI and post-pandemic SCI are the two subgroups of the dataset (SCI) according to the CSIS study. DM methods are employed for feature extraction, and popular ML classifiers including Naïve Bayes (NB), Logistic Regression (LR) Support Vector Machine (SVM), and Random Forest (RF) for classification. Utilizing inputs from six continents worldwide, a centralized classifier technique is employed to preserve a single centralized dataset. To assess this research with greater accuracy and to anticipate which type of SCI can occur in which region of the world, the outcomes of the pre- and post-pandemic datasets were compared. Asia is expected to be the continent hardest impacted by SCI, and it is determined that SVM and RF are significantly superior classifiers than others. Still, the new security risks of the Intelligent CPSS have also become increasingly prominent. Especially in recent years, in Ukraine and Venezuela's power attack incidents, a series of related attacks always occur simultaneously.

Li et al., [13] designed a collection of Cloud-Fog-Edge closed-loop feedback security risk prediction techniques for multi-task compound attacks that combine Markov time-varying systems and categorized deep Boltzmann systems. These techniques are centred on the offensive and defensive concepts of smart games. This approach, which is applicable to many kinds of power intelligent system terminals, combines open interactions, modularity, connectivity, and compliance with standards to foresee security risks. Standardized interfaces utilized to enable interoperability with additional safety devices and provide system security protection features that fit the actual requirements of the industry Internet connection. The procedure is better than the usual standard method, according to the findings. Analysing netflows rather than packets is a recent trend; this method is carried out at a relatively low level, which results in significant false alarm rates.

AlEroud & Karabatis [14] introduced a method that automatically detects and queries potential semantic connections among various kinds of suspicious activity collected from flows utilizing contextual data. Semantic linkages between warnings triggered in consequence of suspicious flows are created by applying time, place, and other contextual data mined from flows. A classifier that examines incoming flows provides an initial prediction to probabilistic SLNs, which will be utilized in an inference process to identify these semantic linkages. Afterwards, run-time queries are made to the SLNs to obtain more pertinent predictions. By treating unknown attacks as versions of known assaults, show how the method might be extended to find unknown attacks within flows.

Mohasseb et al., [15] reported on the findings of a latest case study that examined a community data collection gathered from five Korean small and medium-sized businesses. Cybersecurity incidents and reaction actions are represented by the data set. The research examined problems associated with using attributes taken from previous incidents to forecast reaction actions for upcoming episodes. The evaluation relies on ML techniques for incident classification and response action tracking, together with text mining techniques such n-gram and bag-of-words. Using the findings as a foundation, propose a model for sharing experiences that illustrates how corporations can share their trained classifiers in a collaborative setting without disclosing their individual datasets. For instance, a situation where excellent forecasts are rewarded and unsuccessful ones are penalized is possible imagined utilizing the blockchain system, resulting to a more equitable trading mechanism.

Sarkar et al., [16] attempted to create a system that enables the unusual signals from darkweb forums by utilizing the reply network layout of user interactions in order to forecast external cyberattacks related to

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

enterprises. centered solely on forum, there is an advantage with respect of greater efficiency when considering the reply path framework among groups of users utilizing random walk transitions and communities, according to the model developed for estimating real-world organization cyber-attacks of three distinct security events..

Sentuna et al., [17] proposed an enhanced Naïve Bayes posterior probability (ENBPP) algorithm to increase the cyberattack prediction tools' computing speed and accuracy. A modified risk evaluation function and a modified variant of the NB posterior probability function are combined in the suggested approach. By combining these two features, processing time can be reduced while threat prediction accuracy is increased. To get the outcomes, five separate datasets were utilized.

Alshammari [18] used ML, the issue of scalability in risk assessment may be resolved by using past risk assessments to predict the seriousness of potential future hazards. ML approaches incorporate all of the intuition, insight, and expertise that risk assessors employ to evaluate a danger's severity. ML utilized for assessing risk as a first step. If the risk level is higher than a predefined threshold, further actions can be implemented. Each manual analysis is utilized to train the method, which significantly decreases the requirement for human contact. Obtaining meaningful quantitative estimates of cybersecurity risk variables is challenging, if not impossible. Because risk evaluations in cybersecurity are labor- and time-intensive, they cannot use quantitative calculations.

In their work, Islam et al. [19] introduced an innovative Artificial Intelligence (AI) framework known as SmartValidator. This framework utilizes Machine Learning (ML) techniques to alleviate the workload of human experts and expedite response times, facilitating the automated validation of alerts. SmartValidator is structured into three layers for the execution of data collection, system construction, and alert validation. It approaches the validation process as a classification issue. Rather than creating and storing models for every conceivable requirement, the suggestion is to autonomously generate validation models based on the specific needs of Security Operation Centres (SOC) and Cyber Threat Information (CTI).

Bashir & Arora [20] used Artificial neural network (ANN) for forecasting needs utilizing a survey-based dataset formed up of J&K Union Territory. The dataset employed was developed out of a pressing requirement to ascertain students' knowledge and comprehension levels about cybercrimes. A critical component of preventing cybercrimes against individuals is educating and preparing people for such crimes. To address this issue, an ANN-based system is suggested to forecast when students will require cyber-training. This research suggests using artificial neural networks (ANNs) with backpropagation as a technique to determine whether an individual needs cyber training.

Mall et al., [21] presented a range of DL approaches for effectively identifying DDoS attacks within the SD-CPS architecture, which is based on a flexible and scalable SDN infrastructure. By evaluating a range of DL approaches, the best DL strategies for various assault scenarios can be found. DL models were evaluated on two current security datasets: the SDN-specific dataset and the CICDDoS2019 dataset. They achieved above 99% accuracy in classifying binary and multiclass data across unknown traffic.

Zahra et al., [22] analysed the effect of COVID-19 on several parts of cyber-security and lays out the chronology of worldwide cyberattacks with a Covid-19 topic to determine the attackers' tactics and the effects of their attacks. Additionally, it provides a well-researched set of mitigation techniques that utilized to prevent assaults before they start. Additionally, this publication suggests an intelligence system utilizing fuzzy logic and DM to identify malicious URLs and phishing attempts with a Covid-19 theme. After comparing the system's efficiency to a variety of fraudulent and phishing URLs, it was found that the suggested approach is a workable fix for this issue.

Inference: This study's primary driving force is a centralized classifier that gathers information from all six continents. This method gathers data from six continents and keeps it in a centralized dataset. Additionally, DM, DL methods are well-known ways being utilized to identify cyber security vulnerabilities.

3. Proposed Methodology

This methodology upholds a centralized dataset that includes contributions from six continents. Furthermore, as Fig. 4 illustrates, ODFFLOGNN-based methodologies are well recognized methods for identifying cyber security vulnerabilities. In the proposed ODFFLOGNN model, Logarithmic Neural Layer (LNL) offer a unique

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-255

approach by performing calculations using logarithms within their neurons. The base of the logarithms can be adjusted to suit specific tasks, enhancing model adaptability. ODFFLOGNN integrates the benefits of logarithmic processing with the depth and power of Deep Neural Network (DNN). The HBA is employed to fine-tune hyperparameters, further enhancing model performance.

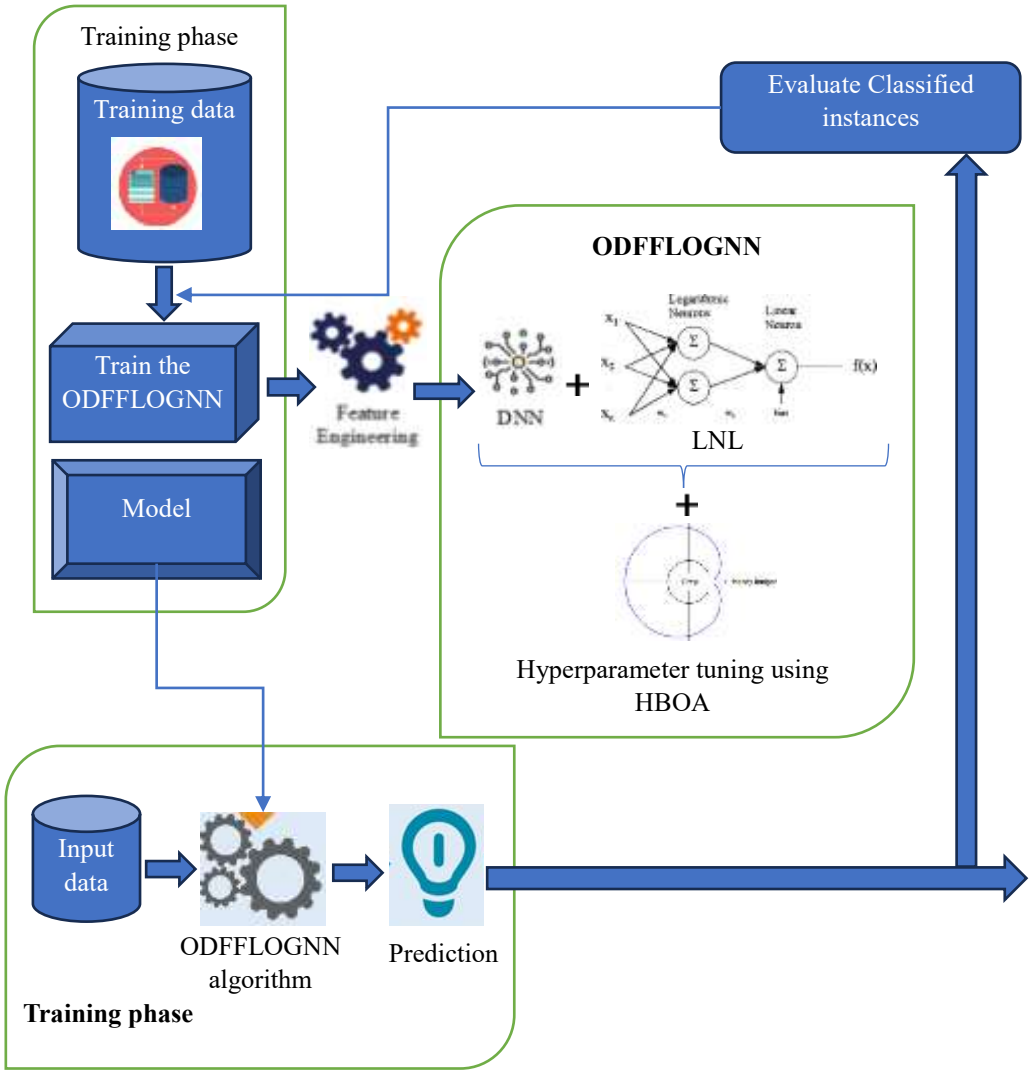

Fig.4. Proposed ODFFLOGNN based SCI detection

3.1. Input Dataset description

- Center for Strategic and International Studies (CSIS) Significant Cyber Incidents dataset [23].
- Content: Includes information on over 1000 SCIs from 2007 to present, covering details like date, target, attack type, and impact.
- Strengths: well-researched, regularly updated, and includes qualitative information like attacker motivation and attribution.

The databases are textual records that provide the type of SCI and the continent on which it originated. It is separated into two sections (pre-pandemic and post-pandemic SCI), after which 10 different types of SCI that happened on six continents were examined. The pre-pandemic (pre-COVID-19) dataset comprises SCI that occurred between 2003 and December 2019. Furthermore, SCI that occurred from January 2020 and the present are included in the post-pandemic (after COVID-19) dataset. Exactly six continents are taken into consideration in this analysis because there are no countries in Antarctica, the seventh continent. Additionally, research is done on ways to employ the data to improve classification accuracy and ultimately create a better classifier for differentiating between SCI. the outcomes attained by concentrating on the kind of SCI that happened on which

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

continent. The goal of this research is to investigate the advantages of centralized classifiers for treating potential SCI.

3.2. Dataset Preprocessing

- **Missing Values Imputation:** Calculate the proportion of missing values in each column based on Mean Imputation. Replaces missing values with the mean of the observed values in that column using Eq. (1)

$$\bar{x} = \frac{(\Sigma x_i)}{n} \quad (1)$$

Where: x̄ is the mean, x_i are the observed values and n is the number of observed values.

- **Inconsistent Formatting:** Identify and standardize inconsistencies in date formats, variable names, etc.

- **Outlier detection statistical measures via Interquartile Range (IQR):** Arrange the data in ascending order. Find the first quartile (Q1), the value that divides the bottom 25% of the data from the top 75%. Find the third quartile (Q3), the value that divides the bottom 75% of the data from the top 25% and calculate the IQR: IQR = Q3 - Q1. Determine outlier boundaries based on lower outlier boundary: Q1 - 1.5 * IQR and Upper outlier boundary: Q3 + 1.5 * IQR. Identify outliers withany data point that falls below the lower outlier boundary or above the upper outlier boundary is considered an outlier.

- **Data Transformation:** Encode categorical variables using one-hot encoding used to transform categorical variables into a numerical format that machine learning algorithms can understand. It involves creating a new binary variable (a dummy variable) for each unique category in the original variable. Let's consider a categorical variable X with K distinct categories as $X = \{x_1, x_2, \ldots, x_k\}$. One-hot encoding creates a new matrix D with K columns, where each column represents a unique category as $D = [d_1, d_2, \ldots, d_k]$. Each element d_{ij} in the matrix is defined as $d_{ij} = 1$ when $x_i =$ j and $d_{ij} = 0$ when $x_i \neq j$.

- **Text Data:** Preprocess text fields using techniques like Tokenization (Split text into individual words or tokens), Stemming or lemmatization (Reduce words to their base forms) and Vectorization (Represent text as numerical vectors (e.g., TF-IDF)).

3.3. Feature Extraction

Data mining characteristics such as n-gram and BoW are often utilized for feature extraction. The uni-gram and bi-gram model concepts were applied in this framework. For a unigram with n=1, for instance, the word "phishing" is extracted from the dataset along with all the SCI that contain it. Additionally, in a bi-gram with n=2, for instance, the two words "SQL Injection" are used to extract the SCI. All uni- and bi-gram words are currently carried by BoW. By applying a filter, this BoW eliminates terms that appear in the dataset with the lowest frequency. The method of text vectorization centered the BoW framework is employed, and these terms weren't utilized any more for features utilizing Term Frequency - Inverse Document Frequency (TF-IDF) (feature vector (504, 6) for pre-pandemic dataset and (543, 6) for post-pandemic dataset).

Classification Using Proposed ODFFLOGNN Model for SCI detection

In the proposed ODFFLOGNN framework, the LNL introduces a unique approach by conducting computations through logarithmic operations within the neural units. The logarithmic base, a crucial parameter, can be tailored to meet specific task requirements, enhancing the adaptability of the model. ODFFLOGNN seamlessly combines the benefits of logarithmic processing with the complexities and computational power inherent in DFFNN. The methodology of HBA is integrated to precisely fine-tune hyperparameters, adding an extra layer of optimization to enhance the overall performance of the model. The following segment offers a thorough explanation of each of the previously mentioned proposed approaches, exploring the workings of their mechanisms in a detailed manner.

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com
DOI https://doi.org/10.15463/gfbm-mib-2023-255

Deep Feed Forward Neural Network (DFFNN) Structure: Fig. 5 shows the standard architecture of a feed-forward DNN. In this image, X stands for the input layer and X_i for the data representation in the i^{th} hidden layer. According to this paradigm, the network consists of v layers, and X_v, the output layer, is responsible for estimating the intended output, Y.

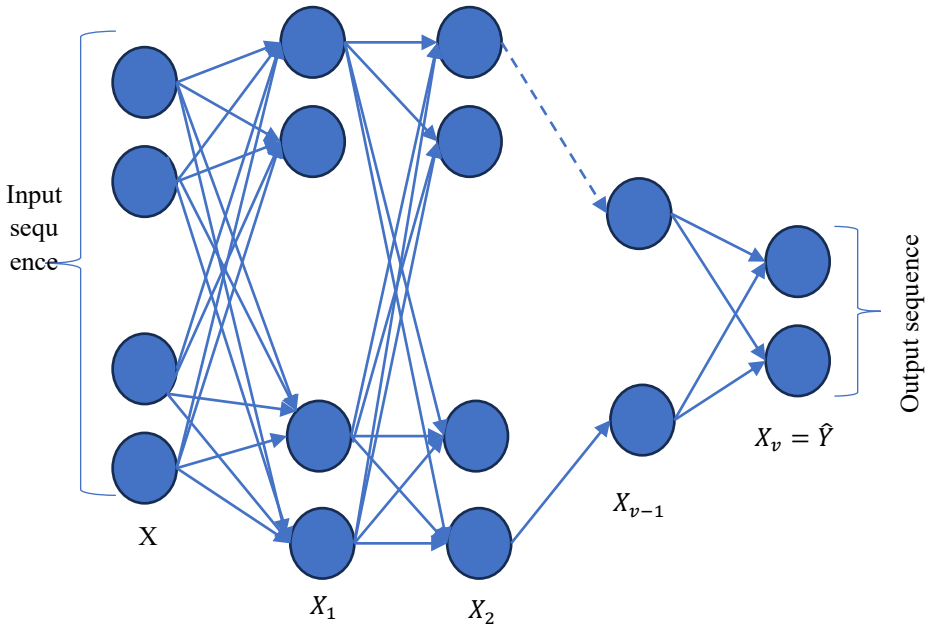

Fig.5. The Structure of DFFNN

The output of the j th neuron in layer n is computed as following. Each layer is formed by several neurons that analyze the data concurrently:

$$x_{n,j} = g_j^n(x_{n-1}) = f^n\left(\sum_{i=1}^{m_{n-1}} \mathbb{w}_{i,j}^n x_{n-1}, + \mathbb{b}_j^n\right) \quad (2)$$

where m_{n-1} is the number of neurons in layer $n-1$ and $\mathbb{w}_{i,j}^n$ is the weight that links the i^{th} neuron's output in layer $n-1$ to the j^{th} neuron's input in layer n. Moreover, the output function of the neurons in layer n is denoted by $f\,n(\cdot)$, and the bias of the j^{th} neuron is represented by $\mathbb{b}_j^n$. To demonstrate (2) utilizing the following vector notation:

$$x_n = G^n(x_{n-1}) \quad (3)$$

where x_n is a neuron outputs combination in layer n, i.e. $x_n = [x_{n,1}, \cdots, x_{n,m_n}]$. k_n is dependent on the input space and the output functions, shows the total number of possible output possibilities in the n^{th} layer. For instance, $k_n = 2^{m_n}$ has binary output functions. $X_n = \{x_1^n, \cdots, x_{k_n}^n\}$ is the set of all potential output possibilities at the n^{th}layer. Remember that the processing power of a single neuron is quite restricted. Activation functions are mathematical operations applied to the output of each neuron in a neural network. They introduce non-linearity to the network, allowing it to learn complex relationships in the data. Without activation functions, a neural network would essentially be a linear model, and stacking multiple linear layers would not increase the network's expressive power. Here use Rectified Linear Unit (ReLU):

$$ReLU(x) = \max(0, x) \quad (4)$$

Essentially, a single neuron is able to execute a hyper-plane in its input space, which means that not every mapping from input to output is possible. The primary causes of neural networks' numerous layers that interact with neurons is this constraint. A mapping among the outputs of the successive levels is carried out by the neural computation that takes place in each layer. It is noteworthy to mention that the logarithmic neuron mapping in a single hidden layer of a DFFNN serves as a segmentation of the layer's input space. Consequently, the function

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

of a DFFNN involves a series of sequential partitioning and mapping processes, all aimed at the estimation of a desired output.

Logarithm Neuron layer in Neural network: Standard DFFNN networks inherently execute addition, subtraction, and multiplication operations involving constants as an integral part of their functioning. A network capable reliably execute multiplications, divisions, and powers over wide ranges of input values is required for non-linear function estimation. Multiplication becomes addition, division becomes subtraction, and powers become multiplication by a constant when one transforms to a logarithmic space. These functions can help capture and amplify small changes in input, which might be useful for detecting subtle patterns or anomalies in data. As an outcome, the network is able to accurately carry out these tasks throughout the whole spectrum of potential inputs. In the logarithmic space, a logarithmic neuron functions. The natural logarithmic space or any other appropriate foundation employed as this space. The input is first logarithmized, after which the modified inputs are multiplied by a weight vector, added together, and subjected to a linear activation function. The output of this function is subsequently subjected to the inverse logarithm. The fundamental logarithm neuron is depicted in Equation 5 and Figure 6, where x represents the input vector, $\mathbb{w}$ denotes the weight vector, and $f(x)$ is the conventional linear activation function.

$$y = \exp\left(f\left(\sum_{i=1}^{n} \mathbb{w}_i \cdot \ln x_i\right)\right) \tag{5}$$

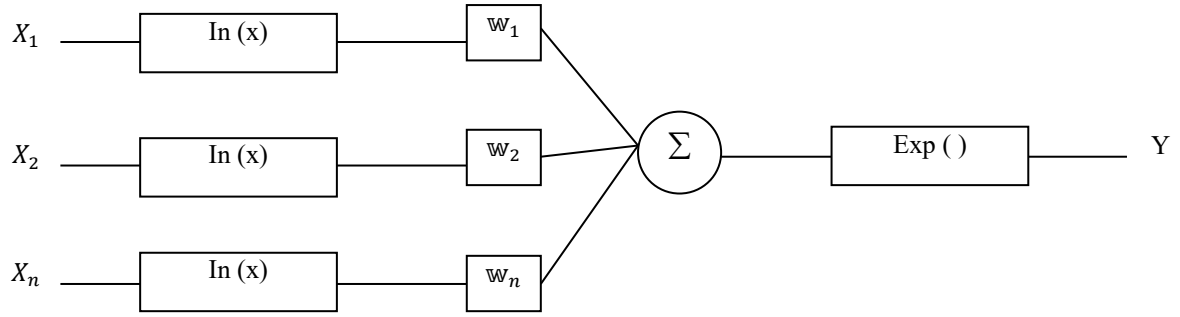

Fig.6. Logarithmic Neuron

The inputs of a single layer logarithmic network are converted to their natural logarithms (ln), and the logarithmic neuron output is where the inverse natural logarithm (ln) is obtained. When addition and subtraction are involved, this approach performs inadequately, but it performs well for the examples that are covered. This issue is resolved by an n-layer DFFNN network, whose output layer is a conventional linear neuron and whose first layer is comprised of logarithmic neurons. The initial layer of the logarithmic network converts the input data into functional representations (FR). These representations are of the form:

$$FR \rightarrow x_1^{\mathbb{w}_1} \ldots x_n^{\mathbb{w}_n} \tag{6}$$

The FR are then combined by the second layer of the network to form the function approximation.

$$f(x) = \mathbb{w}_{21}.FR_1 + \mathbb{w}_{22} \cdot FR_2 + \mathbb{b}ias \tag{7}$$

This function could be approximated by standard DFFNN networks, but it would require a large number of nodes and be limited to a certain range of inputs. For every FR, this network structure just needs one hidden logarithmic neuron, and it works with any input value.

Proposed DFFLOGNN Model: This study presents a formal statement of the suggested logarithmic activation function in Eq. (4), which is one of its main contributions. For unbounded non-linear function approximation, a new design with a logarithmic hidden layer outperforms the conventional DFFNN. This framework utilizes both of the gradient descent training process and a % error goal function. Examples of non-linear function approximations are employed to demonstrate that novel design is more accurate than both the logarithmically converted DFFNN and the standard DFFNN. The logarithmic activation function formed as follows

$$h(x) = \begin{cases} \alpha[log\ (x + c_0) + c_1] & if\ x \geq 0 \\ -\alpha[log\ (c_0 - x) + c_1] & if\ x < 0, \end{cases} \tag{8}$$

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

where $\alpha > 0$ is a tunable scaling parameter, $c_0 = exp(-1)$, $c_1 = 1$, and x is the input. This model defines $c_0$and c_1 To expedite training and prevent a bias shift for subsequent layers, the activation function should be antisymmetric about 0.

As this model employs a gradient-based optimization technique, as deep designs are susceptible to "gradient flow problems," it is imperative to guarantee that the gradient can propagate efficiently throughout the network [24]. There are two possible causes of insufficient gradient flow: either as the gradients propagate throughout the network, they get too tiny (vanishing gradient) or too high (exploding gradient). In both cases, training results in numerical instabilities that render convergence extremely difficult or impossible for large designs. The backpropagation algorithm, which uses the chain rule to calculate gradients, is the cause of this. As a result, every gradient also functions as a multiplication factor for the gradient of the layer above it. The following is the expression for the derivative of Equation (8) with regard to *x* in the given case.

$$\frac{\partial h(x)}{\partial x} = \begin{cases} \frac{\alpha}{x+c_0} & if\ x \geq 0 \\ \frac{\alpha}{c_0 - x} & if\ x < 0 \end{cases} \tag{9}$$

It demonstrates that at zero, the gradient discontinuously changes in size to α/c_0^{-1} as it scales with x^{-1}. The gradient would have a higher chance of disappearing the deeper the design goes if this model applied this activation function to every activation in the network. Furthermore, gradient jumps may result from the discontinuity at $x = 0$, which would further reduce numerical stability. Consequently, this framework employs a Relu activation on all hidden layers and a logarithmic activation function solely on the first convolutional layer. Without sacrificing numerical stability, this tradeoff preserves the diffraction pictures' exponential scale. Because α is an adjustable hyperparameter, this framework tests three values for $\alpha \in [0.2, 0.5, 1.0]$ and assesses the effect on the neural network's efficiency. The deep neural network's adaptability and general efficiency are enhanced by the logarithmic activation function. This is expected as the framework leveraged a known feature of the dataset to apply a form of feature engineering to the network. Consequently, without escalating the complexity, depth, the model demonstrated that incorporating the logarithmic activation enhances all pertinent evaluation metrics. Hence, the model adopts the logarithmic activation function with a default α value of 0.2 for all subsequent experiments.

DFFNNs are susceptible to overfitting, particularly in scenarios involving limited training data. Moreover, due to their large parameter count, DFFNNs have a notable ability to memorize intricate details within the training data, including noise and outliers. This circumstance may lead to a model excelling on the training set but struggling to generalize effectively to unfamiliar, unseen data. Hyperparameter tuning plays a pivotal role in the training of DFFNN as it enables the optimization of model performance through the identification of the most effective combination of hyperparameter values. DNNs are intricate models with a multitude of hyperparameters, and the selection of these parameters can profoundly influence the model's capacity to learn and generalize from the provided data.

Hyperparameter tuning using honey badger Algorithm: The vast, weasel-like honey badger inhabits the semi-desert regions of Africa, Southwest Asia, and the Indian subcontinent. It has fluffy black and white hair. The HBA is a system designed to simulate the intelligent hunting behavior of honey badgers, which locate food by either following the honeyguide bird or sniffing and digging. The "digging phase" and the "honey phase" are the two stages that make up the HBA method. As a result, it employs its sense of smell to determine the approximate location of the prey during the first phase, known as the digging mode. When it gets close to the prey, it shifts positions to determine the optimum spot to dig and seize it. In a subsequent stage known as the honey mode, the honey badger employs the honeyguide bird as its primary route to the beehive. HBA can potentially optimize DFFLOGNN architecture (number of layers, neurons, weight and bias) alongside hyperparameters. The pseudocode of HBA based hyperparameter tuning of DFFLOGNN is given in Table 1. This provides a quantitative explanation for these procedures [25]. Fitness Function defined as follows:

$$f(z_k) = Evaluation\big(DFFLOGNN(X_i)\big) \tag{10}$$

The HBA evaluates each solution z_k by training a DFFLOGNN with its encoded hyperparameters and measuring its performance on a validation set using a problem-specific metric (accuracy).

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

Initialization phase. The equation following utilized to determine the honey badgers' beginning population size (PS) and locations:

$$z_k = lo\mathscr{b}_k + rand_1 \times (up\mathscr{b}_k - lo\mathscr{b}_k); k = 1, 2, \ldots, PS \quad (11)$$

where z_k signifies the k^{th} honey badger in a population of PS objects (also z_k represents the k-th solution in the HBA population, encoding a set of DFFLOGNN hyperparameters). here, $up\mathscr{b}_k$ and $lo\mathscr{b}_k$ signify the lower and upper boundaries of the search space. $rand_1$ is a random number in the range [0,1].

Defining intensity (I). This step's goal is to calculate the intensity centered on the prey's attention force and the badger's closeness to the prey, k^{th}. According to this equation, the prey will move quickly if its scent intensity (SI_k) is high. According to the inverse square law, if the fragrance intensity $SI_k$is low, it will move slowly, as indicated by the following equation:

$$SI_k = rand_2 \times \frac{AF}{4\pi d_k^2}, AF = (z_k + z_{k+1})^2, dist_k = z_{prey} - z_k \quad (12)$$

where AF is the attention force, $dist_k$ signifies the distance of the prey from the k^{th} honey badger, and z_{prey} is the prey's position. $rand_2$ is random value in the range of [0, 1].

Update density factor (∂). This stage manages the time-based modifications' randomization to enable a seamless transfer from exploration to exploitation. In this context, modify the density factor (∂) by decreasing it in each iteration, aiming to diminish the likelihood of randomization over time. Utilize the provided equation for this adjustment:

$$\partial = cn \times exp\left(\frac{-t}{t_{max}}\right) \quad (13)$$

where t signifies current iteration number, cn is a constant number equal to 2, and t_{max} signifies the maximum number of iterations.

Escaping and averting local optimum. The goal of this phase and the next two is to break out of the best local positions. To give agents and candidates more opportunities to completely search the search space, the HBA in this instance employs a flag named F to change the search direction.

Updating the positions of agents/candidates. The two stages of the HBA position update procedure (z_{new}) were discussed. The "honey phase" and the "digging phase" are the names given to these stages. The clarification in further detail can be found below.

Digging phase. After locating their meal with their acute sense of smell, honey badgers use their cardioid shape to dig a hole around it. During the excavation stage, the HBA update position operation (z_{new}) is executed:

$$z_{new} = z_{prey} + E \times \gamma \times SI \times z_{prey} + F \times rand_3 \times \partial \times dist_k \times |\cos(2\pi rand_4) \times [1 - \cos(2\pi rand_5)]| \quad (14)$$

Where z_{prey} signifies the finest estimation of the prey's place, F is flag, SI signifies the intensity stated in step 2, and as stated in step 3, the update density factor is represented by ∂. The variable $\gamma \geq 1$ indicates the honey badger's ability to gather food; it is set to 6 by default. The disk measures the honey badger's distance from its target. "| |" stands for the absolute operator, and the random numbers in the interval [0,1] are shown by the symbols $rand_3, rand_4$, and $rand_5$. The formula represents the flag F, which is employed to change the search direction:

$$z_{new} = z_{prey} + F \times rand_6 \times \partial \times dis_k \quad (15)$$

where z_{new} specifies the honey badger's new position, and z_{prey} indicates the prey's position. $rand_6$ is random value in the range [0, 1]. Equations (13) and (14) utilized to calculate ∂ and E, individually. Equation (15) shows that, according to the knowledge gathered about the distance dis_k, the honey badger searches close to the prey's location z_{prey} because it is currently detected. At this point, the search is impacted by the time-varying search behavior ∂. A honey badger might find a F disturbance. The main key function of ODFFLOGNN is given as follows:

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

- InitializePopulation: Creates initial HBA solutions with random hyperparameter values within specified ranges.
- EvaluateDFFLOGNN: Trains a DFFLOGNN with given hyperparameters and evaluates its performance on a validation set.
- SelectMode: Determines whether to apply digging or honey mode based on a transition probability function.
- DiggingMode: Explores the search space, updating solutions based on digging behavior equations.
- HoneyMode: Exploits promising solutions, refining them using honey mode equations.

Table 1. The pseudocode of HBA based hyperparameter tuning of DFFLOGNN

```
Input: HBA_DFFLOGNN_Tuning (DFFLOGNN_model, hyperparameter_ranges, max_iterations)
Output: hyperparameters of DFFLOGNN
    # Initialize population of HBA solutions
    population = InitializePopulation(hyperparameter_ranges)
    best_solution = None
    best_fitness = -inf
    for iteration in range(max_iterations):
        # Evaluate fitness of each solution
        for solution in population:
            fitness = EvaluateDFFLOGNN(DFFLOGNN_model, solution)
            solution.fitness = fitness
            # Update best solution if necessary
            if fitness > best_fitness:
                best_solution = solution
                best_fitness = fitness
        # Apply digging mode or honey mode based on transition probability
        mode = SelectMode(iteration, max_iterations)
        if mode == "digging":
            population = DiggingMode(population)
        else:
            population = HoneyMode(population)
    # Return the best hyperparameter configuration found
    return best_solution
function EvaluateDFFLOGNN(DFFLOGNN_model, hyperparameters):
    # Set DFFLOGNN hyperparameters according to the solution
    SetHyperparameters(DFFLOGNN_model, hyperparameters)
    # Train the DFFLOGNN model
    TrainDFFLOGNN(DFFLOGNN_model)
    # Evaluate performance on a validation set
    performance = EvaluatePerformance(DFFLOGNN_model)
    return performance
```

4. Experimental Results and Discussion

This study examines four distinct classifier types: SVM, NB, LR, and the suggested ODFFLOGNN. An experimental investigation is conducted to examine the classifier's capabilities. Depending on the kind of SCI, the classifier's output predicts the name of the continent. Accuracy, Recall, Precision, and F1-score are employed as performance indicators to assess the classifiers' efficiency. The system undergoes 1047 SCI training to assess the classifiers' efficiency.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100 \quad (15)$$

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com
DOI https://doi.org/10.15463/gfbm-mib-2023-255

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \times 100 \quad (16)$$

$$\text{F1} - \text{score} = 2 * \left(\frac{\text{precision} * \text{recall}}{\text{precision+recall}}\right) \quad (17)$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \times 100 \quad (18)$$

False Negative (FN) represents the # of actual attack cases that are mistakenly categorized as normal, True Positive (TP) represents the # of actual attack records that are properly identified as attacks, True Negative (TN) represents the # of actual normal data that has been identified as normal, and False Positive (FP) represents the # of actual typical events that are mistakenly identified as attacks.

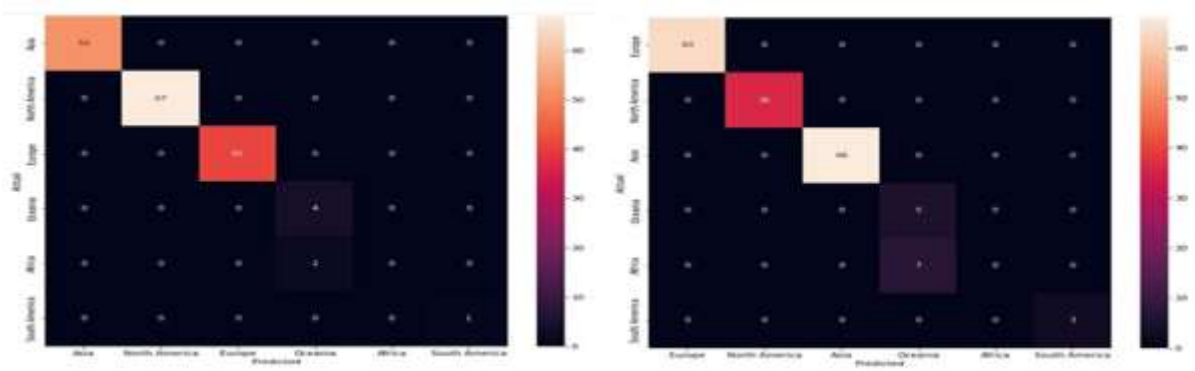

Fig.7. Confusion matrix for proposed ODFFLOGNN (Left: pre-pandemic dataset. Right: post-pandemic dataset).

The classifier's quality is demonstrated by its good precision, recall, and F1 measures compared to Asia, Europe, North America, and South America, all of which are around 99.58. Various assessment metrics are employed to forecast the research's correctness. Similar outcomes are seen in Fig. 7's Confusion matrix, which distinguishes between real and expected outcomes. The real and predicted values for each continent are displayed in the ODFFLOGNN classifier confusion matrix. Table 2 displays the precision values. For example, the values of the Asia continent revealed 52 for a real case and 66 for a post-pandemic case for the actual and identical values for the projected. In the assessment evaluate, all the values from the other continents can be compared and verified. The research's heatmap, which is displayed in Fig. 7, compares the accuracy level to real and anticipated values.

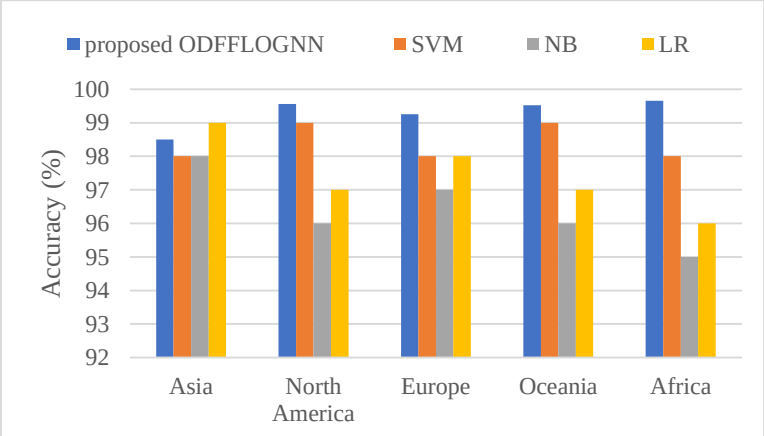

Fig 8: Accuracy performance comparison of SCI detection

When considering the total amount of times in a given database, Fig. 8 compares the accuracy of suggested ODFFLOGNN with Compared to all other models like SVM, NB and LR achieves the accuracy of five categories such as Asia of 98.5%, North America of 99.56%, Europe of 99.25%, Oceania of 99.52%, Africa of 99.65% and South Africa of 99.58%. This is because it does not require a sizable number of derived components

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

during reduction. Consequently, the new method ODFFLOGNN outperforms current algorithms in terms of improved validation results for identifying SCI. Training SVMs, especially with large datasets, can be computationally expensive. NB assumes features are independent of each other, which may not hold true in complex cyber security scenarios. Visualizing decision boundaries or using advanced interpretability techniques might be necessary to gain deeper understanding of the model's predictions. This can limit its practical application in real-time cyber incident detection scenarios. ODFFLOGNNs can capture intricate, non-linear relationships between features, crucial for detecting complex cyber-attacks, where logarithmic activation functions can compress feature space, reducing computational costs and potentially improving generalization. The numerical results of Accuracy performance comparison of SCI detection are shown in Table 2.

Table 2. The numerical results of Accuracy performance comparison of SCI detection

SCI Class	proposed ODFFLOGNN	SVM	NB	LR
Asia	98.5	98	98	99
North America	99.56	99	96	97
Europe	99.25	98	97	98
Oceania	99.52	99	96	97
Africa	99.65	98	95	96
South Africa	99.58	99	97	98

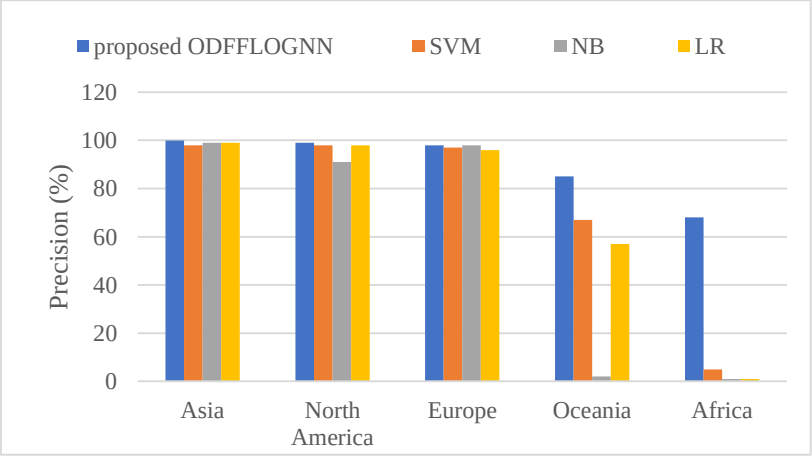

Fig 9: Precision performance comparison of SCI detection

The precision of the suggested ODFFLOGNN and current algorithms for the total number of times in a certain database is shown in Fig. 9 and the results are compared with other models like SVM, NB and LR. It shows the ODFFLOGNN to current algorithms; excellent accuracy was achieved. In comparison to all other models the proposed scheme achieves the precision of five categories such as Asia of 98.7%, North America of 98.5%, Europe of 98.6%, Oceania of 98.3% Africa of 98% and South Africa of 98.8%. The performance of SVMs is heavily influenced by the choice of kernel. Finding the optimal kernel for a specific dataset can involve experimentation and domain knowledge. Similar to SVMs, NB can be sensitive to feature scaling and require appropriate normalization for optimal performance. This is because it does not require many derived components during reductions. LR can struggle with imbalanced datasets, potentially missing rare but significant attacks. Regarding improved validation findings for identifying SCI, the suggested method ODFFLOGNN is superior to the current methods. The logarithmic layer might improve feature extraction and representation, aiding DNN's ability to detect subtle attack patterns and thus improves the precision results. They automatically extract meaningful features from raw data, reducing manual feature engineering efforts. The numerical results of precision performance comparison of SCI detection are shown in Table 3.

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

Table 3. The numerical results of precision performance comparison of SCI detection

SCI Class	Proposed ODFFLOGNN	SVM	NB	LR
Asia	99.89	98	99	99
North America	99	98	91	98
Europe	98	97	98	96
Oceania	85	67	2	57
Africa	68	05	1	1
South Africa	99	98	4	98

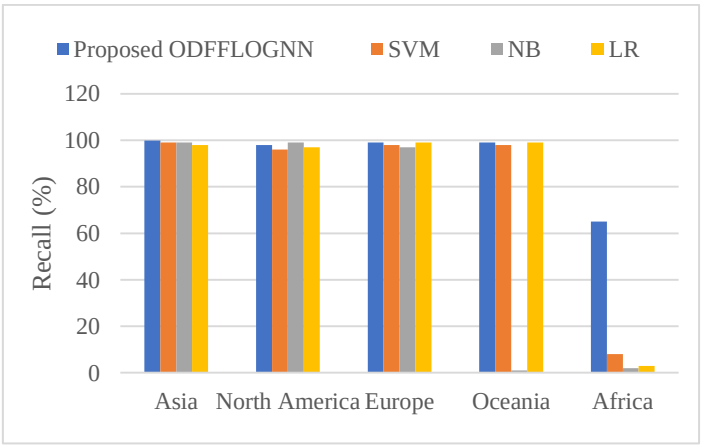

Fig 10: Recall the performance comparison of SCI detection

Fig. 10 compares the Recall of current models and suggested ODFFLOGNN models. Comparing the ODFFLOGNN to current algorithms like SVM, NB and LR, excellent accuracy was achieved. The ODFFLOGNN attains recall rates of five divisions such as Asia of 99.89%, North America of 98%, Europe of 99%, Oceania of 99% Africa of 65% and South Africa of 99.89%, contrasted to all the other algorithms like other algorithms, with these processes significantly boosting the quality of ODFFLOGNN hyperparameters with less computational expenses and thus achieving high Recall. SVMs can face challenges in effectively handling high-dimensional data, which is common in cybersecurity domains. This can lead to overfitting and poor generalization. NB struggles to capture complex relationships between features, potentially missing subtle patterns indicative of real attacks. Similar to NB and SVM, feature scaling is crucial for optimal performance. ODFFLOGNNs have the ability to learn and adapt to evolving attack patterns, making them well-suited for continuous monitoring. They can effectively handle significant variations in feature values, a common occurrence in cybersecurity data. Regarding improved validation findings for identifying SCI, the suggested method is superior to the current methods. The numerical results of Recall performance comparison of SCI detection are shown in Table 4.

Table 4. The numerical results of Recall performance comparison of SCI detection

SCI Class	Proposed ODFFLOGNN	SVM	NB	LR
Asia	99.89	99	99	98
North America	98	96	99	97

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-255

Europe	99	98	97	99
Oceania	99	98	1	99
Africa	65	8	2	3
South Africa	99.89	99	3	99

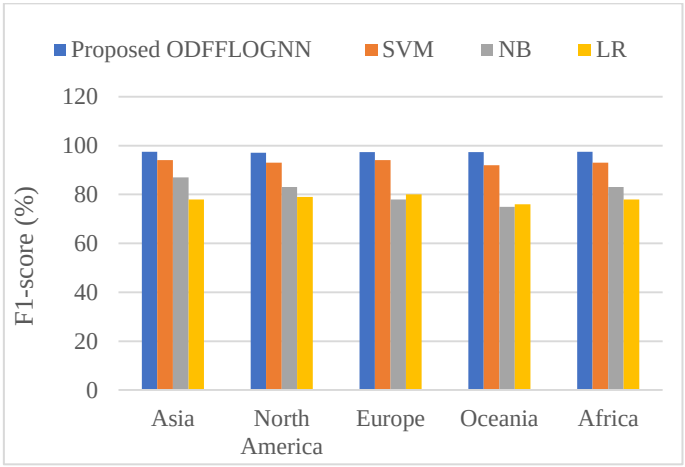

Fig 11: F1-score performance comparison of SCI detection

The suggested ODFFLOGNN and current models' F1-scores for the number of epochs in a particular database are shown in Fig. 11. Comparing the ODFFLOGNN to other models like SVM, NB and LR; excellent accuracy was achieved. In comparison to all of the other algorithms, the ODFFLOGNN achieves the precision of five classes, such as Asia of 99.78%, North America of 98%, Europe of 99%, Oceania of 89% Africa of 64% and South Africa of 99%. In cybersecurity, the distribution of classes (normal vs. anomalous events) is often imbalanced. SVMs can be biased towards the majority class, leading to missed detection of rare but significant incidents. NB might not adapt well to evolving cyber threats and require frequent retraining as attack patterns change. Compared to NB, LR training can be computationally more expensive, especially with large datasets. They can handle large, high-dimensional datasets common in cybersecurity, the logarithmic layer might improve feature extraction and representation, aiding DNN's ability to detect subtle attack patterns. With these algorithms, the quality of the ODFFLOGNN hyperparameters is significant. The numerical results of f1-score performance comparison of SCI detection are shown in Table 5.

Table 5. The numerical results of f1-score performance comparison of SCI detection

SCI Class	**Proposed ODFFLOGNN**	**SVM**	**NB**	**LR**
Asia	99.78	99	98	98
North America	98	96	95	99
Europe	99	97	99	98
Oceania	89	80	4	70
Africa	64	4	3	2
South Africa	99	96	1	98

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

5. Conclusion and future work

According to the CSIS study, this effort emphasizes SCI research conducted between September 2003 and October 2022. DL and DM techniques are employed to examine and categorize the datasets. Three distinct classifiers, namely NB, SVM, and LR, are evaluated with the suggested ODFFLOGNN and utilized to predict the output (the name of the continent depending on the type of SCI). Additionally, the continent most likely to experience SCI during that time is forecasted. Asia is the continent most afflicted by SCI, and ODFFLOGNN is ultimately found to be superior to other classification methods in both pre- and post-pandemic settings. Honey Badger's ability to handle complex landscapes might be advantageous for imbalanced datasets, where attacks are rare. It could find optimal hyperparameters that address class imbalance issues, improving detection of less frequent but significant attacks. From the experimental results, it is concluded that the accuracy of suggested ODFFLOGNN achieves the accuracy of five categories such as Asia of 98.5%, North America of 99.56%, Europe of 99.25%, Oceania of 99.52%, Africa of 99.65% and South Africa of 99.58%. In future work, explore federated learning approaches with optimization technique where models are trained on decentralized data without sacrificing privacy. This is crucial for sharing cyber threat intelligence without compromising sensitive data. Develop privacy-preserving deep learning techniques for cyber incident detection that protect individual data while preserving the overall effectiveness of the system.

References

1. European Cyber Security Organisation. https://www.ecs-org.eu/. Accessed 01 July 2018
2. Bahraminikoo P, Yeganeh M, Babu G (2012) Utilization data mining to detect spyware. IOSR J Comput Eng 4(3):01–04
3. Yu, Z., Kening, G., Feng, L., & Ge, Y. (2014, September). A new method for link prediction using various features in social networks. In *2014 11thThis model b Information System and Application Conference* (pp. 144-147). IEEE.
4. Fenz, S., Heurix, J., Neubauer, T., & Pechstein, F. (2014). Current challenges in information security risk management. *Information Management & Computer Security*, *22*(5), 410-430.
5. Liu, Y., Sarabi, A., Zhang, J., Naghizadeh, P., Karir, M., Bailey, M., & Liu, M. (2015). Cloudy with a chance of breach: Forecasting cyber security incidents. In *24th USENIX Security Symposium (USENIX Security 15)* (pp. 1009-1024).
6. Sconzo, M. (2017). SecRepo. com-samples of security related data. *Last accessed*, *5*.
7. Center for Applied Internet Data Analysis: CAIDA Data. http://www.caida.org/data/overview/. Accessed 05 Jan 2017
8. The Anatomy of the SolarWinds Attack Chain (cyberark.com)
9. Analyzing attacks taking advantage of the Exchange Server vulnerabilities | Microsoft Security Blog
10. Cyber-attacks during the Russian invasion of Ukraine - Office for Budget Responsibility (obr.uk)
11. Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, *18*(2), 1153-1176.
12. Mumtaz, G., Akram, S., Iqbal, W., Ashraf, M. U., Almarhabi, K. A., Alghamdi, A. M., & Bahaddad, A. A. (2023). Classification and Prediction of Significant Cyber Incidents (SCI) using Data Mining and Machine Learning (DM-ML). *IEEE Access*.
13. Li, Q., Tian, Y., Wu, Q., Cao, Q., Shen, H., & Long, H. (2020). A cloud-fog-edge closed-loop feedback security risk prediction method. *IEEE Access*, *8*, 29004-29020.
14. AlEroud, A. F., & Karabatis, G. (2016). Queryable semantics to detect cyber-attacks: A flow-based detection approach. *IEEE transactions on systems, man, and cybernetics: systems*, *48*(2), 207-223.
15. Mohasseb, A., Aziz, B., Jung, J., & Lee, J. (2020). Cyber security incidents analysis and classification in a case study of Korean enterprises. *Knowledge and Information Systems*, *62*, 2917-2935.
16. Sarkar, S., Almukaynizi, M., Shakarian, J., & Shakarian, P. (2019). Mining user interaction patterns in the darkweb to predict enterprise cyber incidents. *Social Network Analysis and Mining*, *9*, 1-28.
17. Sentuna, A., Alsadoon, A., Prasad, P. W. C., Saadeh, M., & Alsadoon, O. H. (2021). A novel Enhanced Naïve Bayes Posterior Probability (ENBPP) using machine learning: Cyber threat analysis. *Neural Processing Letters*, *53*, 177-209.

ISSN: 0937-583x Volume 88, Issue 10 (Nov -2023)
https://musikinbayern.com DOI https://doi.org/10.15463/gfbm-mib-2023-*255*

18. Alshammari, F. H. (2023). Design of capability maturity model integration with cybersecurity risk severity complex prediction using bayesian-based machine learning models. *Service Oriented Computing and Applications*, *17*(1), 59-72.
19. Islam, C., Babar, M. A., Croft, R., & Janicke, H. (2022). SmartValidator: A framework for automatic identification and classification of cyber threat data. *Journal of Network and Computer Applications*, *202*, 103370.
20. Bashir, S., & Arora, B. (2023). Prediction of need for cyber training for university students using artificial neural networks. *Procedia Computer Science*, *218*, 1414-1423.
21. Mall, R., Abhishek, K., Manimurugan, S., Shankar, A., & Kumar, A. (2023). Stacking ensemble approach for DDoS attack detection in software-defined cyber–physical systems. *Computers and Electrical Engineering*, *107*, 108635.
22. Zahra, S. R., Chishti, M. A., Baba, A. I., & Wu, F. (2022). Detecting Covid-19 chaos driven phishing/malicious URL attacks by a fuzzy logic and data mining based intelligence system. *Egyptian Informatics Journal*, *23*(2), 197-214.
23. Significant Cyber Incidents (SCIs). [Online]. Available: https://www. csis.org/programs/strategic-technologies-program/significant-cyberincidents.
24. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).
25. Manzhos, S., Yamashita, K., & Carrington Jr, T. (2009). Fitting sparse multidimensional data with low-dimensional terms. *Computer Physics Communications*, *180*(10), 2002-2012.