A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

Golla jyothirmai  
Both UMKC, Kansas, Missouri, USA  
jyothirmai100@gmail.com

Abstract: The issue of class imbalance, on the other hand, has yet to be resolved. In this research work, an intrusion detection model for class imbalance data is introduced by following four major phases: (a) pre-processing, (b) imbalance processing, (c) feature extraction, and (d) intrusion detection phase. The overall architecture of the proposed work is manifested in Fig.1. Initially, the collected class imbalance data is pre-processed via data cleaning and a data standardization approach. Then, the pre-processed class imbalance data is balanced by an improved over-sampling technique using SMOTE and the Multi-kernel FCM clustering model. Subsequently, multi-features like Exponential Moving Averages (EMA), Double Exponential Moving Averages (DEMA), Improved weighted holoentropy (IWH), and statistical features (mean, median, standard deviation, percentile, and moment) are extracted from these balanced data. The extracted overall features are given for K-fold validation. Then, the intrusion detection phase is modeled with a combination of four individual machine learning models, such as Support Vector Machine (SVM), Deep Belief Network (DBN) with incremental learning, Naïve Bayesian (NB), and Random forest (RF) to achieve the utmost detection performance. Each of the classifiers is trained with the acquired K-fold data features. Moreover, to achieve the utmost detection accuracy and better tradeoff performance, the weight of DBN is fine-tuned by a new Self improved Seagull optimization algorithm (SI-SOA), which is an improved version of the standard Seagull optimization algorithm (SOA). Then, the logoff performance is computed from the outcome acquired from each of the individual classifiers (SVM, DBN, NB, and RF). Finally, the log computed result will portray the detected attacks in the network.

Keywords: Intrusion Detection; Imbalanced Dataset; Improved Weighted Holoentropy (IWH); Incremental Learning; Optimized DBN; SI-SOA.

Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMOTE</td>
<td>Synthetic Minority Over-Sampling Technique</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Characteristics Operating</td>
</tr>
<tr>
<td>GAN</td>
<td>Generative Adversarial Network</td>
</tr>
<tr>
<td>FNN</td>
<td>Feed-forward Neural Network</td>
</tr>
<tr>
<td>A-NIDS</td>
<td>Anomaly-based Network Intrusion Detection System</td>
</tr>
<tr>
<td>b-XGBoost</td>
<td>binary eXtreme Gradient Boosting</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communications Technology</td>
</tr>
<tr>
<td>IGAN</td>
<td>Imbalanced Generative Adversarial Network</td>
</tr>
<tr>
<td>DNN</td>
<td>Deep Neural Network</td>
</tr>
<tr>
<td>EMA</td>
<td>Exponential Moving Averages</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection Systems</td>
</tr>
<tr>
<td>DM</td>
<td>Data Mining</td>
</tr>
<tr>
<td>IWH</td>
<td>Improved Weighted Holoentropy</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>DBN</td>
<td>Deep Belief Network</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>DEMA</td>
<td>Double Exponential Moving Averages</td>
</tr>
<tr>
<td>NB</td>
<td>Naïve Bayesian</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>MA</td>
<td>Moving Average</td>
</tr>
<tr>
<td>DNN</td>
<td>Deep Neural Network</td>
</tr>
<tr>
<td>FCM</td>
<td>Fuzzy C-Means</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under Curve</td>
</tr>
<tr>
<td>U2R</td>
<td>User To Root</td>
</tr>
</tbody>
</table>
1. Introduction

Many small and large businesses have been using internet services to carry out their everyday operations in recent decades [1]. The creation of applications operating on multiple platforms provides an impetus to the security of the web server, despite the immense rise in the use of computers via networks and information sharing. Malicious cyberattacks on web servers are becoming incredibly popular. Nevertheless, due to its immense volume, physically monitoring network traffic seems to be impractical. Generally, an attack is defined as any access to the network that compromises the data's availability, privacy, or consistency [2] [3], [4] [5]. The number of security risks gets raised as the number of internet applications expands [6] [7] [8] [9] [10]. IDS [4][5] is indeed a system that automatically monitors network traffic to distinguish malicious assaults from regular traffic. ML algorithms can be used to efficiently process large amounts of data. The benchmark dataset NSL KDD [6] [11] [12] is used to evaluate IDSs. A classification problem may well be described as the procedure of accurately recognizing the intrusions from network traffic. A classification method must be designed to successfully detect intrusions with fewer false alarms and increase detection performance [21][22] [23] [24].

Many researchers suggested many ML methodologies and techniques for formulating an efficient IDS, including DT [7], SVM [8], AdaBoost [9], KNN [10], NB [11], and so on. When compared to normal traffic, the number of attacks is a bit less. Conventional classifiers are ineffective in identifying the sorts of attacks. A class Imbalance Problem occurs when the quantity of regular traffic exceeds that of attack traffic. [13], [14], [15], [16], [17]. Because of their low representation in comparison to the majority class (normal class), the instances belonging to the class of interest, i.e., the minority class (attack class), are frequently overlooked [25], [26], [27], [28]. Despite the great accuracy of this technique, the IDS is defunct without effective protection against attack traffic [18], [19], [20]. Hence, specialized techniques, which provide due importance to the minority class are required.

1.1 Data Level Approach for dealing with unbalanced data: Resampling Techniques

When dealing with unbalanced datasets, techniques such as enhancing classification algorithms or balancing classes throughout the training data (data preparation) must be used even before data is fed into the ML techniques. The latter method is recommended since it is more widely applicable. The basic goal of class balancing would be to increase the underrepresented class's occurrence while lowering the dominant class's occurrence. This one is done to ensure that each classes have a roughly equal number of incidents. The following resampling methods are commonly utilized:

1.1.1 Random Under-Sampling:

By randomly removing the majority class instances, Random Under sampling tries to balance the class distribution. This process is repeated until the overwhelming as well as minority class occurrences become equal.

Advantages

When the training data set is large, this can assist reduce run time and storage concerns by decreasing the amount of training data samples.

1.1.2 Random Over-Sampling

Over-sampling is a technique for increasing the number of examples in the minority class by randomly repeating them to offer a more representative sample of the minority class.
Advantages
Unlike sampling, this approach does not result in information loss

1.1.3 Cluster-Based Over Sampling
The K-means clustering methodology is used for minorities and dominant class instances independently. This is done to find clusters inside the data. Following that, each cluster is oversampled to ensure that all clusters have the same class and instances with the same size.

Advantages
- This clustering approach aids in overcoming the issue of class imbalance. Where the number of positive class examples differs from the number of negative class instances.
- Overcome obstacles inside class imbalance, when a class is made up of many sub-clusters. And there aren't the same amount of instances in each sub-cluster.

1.1.4 Informed Over Sampling: SMOTE for imbalanced data
This is utilized to overcome the over-fitting issue. As an example, a sample of information from the minority class is collected, and new synthetic comparable cases are generated. The original dataset would then be supplemented with these synthetic examples. The fresh dataset is used to train the classification techniques as a sample.

Advantages
- Because synthetic examples are created rather than replications of real cases, the problem of overfitting induced by random oversampling is mitigated.
- There will be no information loss.
- In the current research work, the Data Level approaches for class imbalance problems are put together, therefore the proposed work becomes much more suitable to overcome the class imbalance problems in the Intrusion Detection.
- The major contribution of this research work is:
- Improved weighted holoentropy is also determined along with the statistical and other technical indicators.
- Introduces an optimized DBN model to detect the presence of intruders in the network, where the weights are optimally tuned by a new improved algorithm.
- Proposes a SI-SOA for solving the optimization issue.

The remaining section of this paper is organized as: Section 2 addresses the recent works undergone in literature regarding this subject. Section 3, Section 4 and Section 5 depict about proposed IDS in the imbalanced dataset, proposed IDS in the imbalanced dataset, and imbalance processing: improved over sampling technique with SMOTE and multi-kernel FCM clustering model respectively. The multi-feature extraction, k-fold validation, and intrusion detection phase are portrayed in Section 6, Section 7, and Section 8. The results acquired with the projected model are discussed comprehensively in Section 9. This paper is concluded in Section 10.

2. Literature Review

2.1 Related Works
In 2020, Zhang et al. [1] proposed a class imbalance processing technology for the large-scale dataset, referred to as SGM, which combines SMOTE and under-sampling for clustering based on GMM. Researchers subsequently developed SGM-CNN, a flow-based intrusion detection model that incorporates unbalanced class processing with a CNN, and examined how varying amounts of convolution kernels and learning rates affect the performance of the model. The UNSW-NB15 and CICIDS2017 datasets were used to verify the effectiveness of the suggested model.

In 2022, Jiyuan Cui et al. [46] have based on GMM-WGAN-IDS. This method improved the accuracy of minority class detection and reduced false alarm rates in network data that had high dimensions, complexity, and imbalance issues. There were three parts: the SAE, GMM-WGAN, and CNN LSTM. By stacking numerous auto encoders, SAE was able to extract the best low-dimensional characteristics from high-dimensional data. GMM-WGAN reduced the imbalances between the majority class under sampling in the minority class generation and the GMM clustering algorithm using the same model's Wasserstein GAN technique. CNN LSTMs were used to maximize performance by merging CNN and LSTM networks.
for the classification module. Experiments were conducted on the UNSW NB15 and NSL KDD datasets, showing better results than existing methods, and confirming the effectiveness of the system.

In 2022, Balyan et al. [47] focused on developing an effective IDS system. Due to the rapid development of IT technology, these IDS utilized ML techniques to track down suspicious network behavioral patterns and identify harmful intrusions in digital data. To deal with the issue of a limited training dataset, this research enhanced the genetic algorithm and particle swarm optimization (EGA-PSO), as well as improved RF. EGA-PSO helped enhance minor datasets by selecting the best features for better fitness outcomes, while IRF eliminated less significant attributes from each iterative process. Performance tests were conducted on NSL_KDD benchmark datasets, where the HNIDS achieved higher accuracy than other ML models such as SVM, RF LR, etc.

In 2022, Le et al. [48] have introduced the advanced digital technology and IIoT. It also talked about the IDS, a safety solution that helped to detect abnormal behavior in networks to avoid attacks. ML-based IDS models were used for building secure applications, but they suffered from reduced accuracy due to imbalanced multiclass IIoT datasets. This used the highly gradient boosting model to run an IDS on two contemporary imbalanced IIoT datasets, TON IOT, and X-IIoTDS, with outstanding attack detection results attained by F1 scores.

In 2021, Andresini et al. [2] utilized a DL methodology for the binary classification of network traffic. The fundamental concept was to describe network flows as 2D pictures and train a GAN and a CNN using this imagery representation of network traffic. By supplementing the training data necessary to develop a CNN-based intrusion detection model, the GAN has been trained to provide fresh images of unexpected network intrusions. Furthermore, GAN-based data augmentation was permitted to focus on the potential unbalance of malicious activity in network traffic, which has been often more uncommon than regular traffic. It’s being used to develop a strong intrusion detection model by simulating unexpected assaults. On four benchmark datasets, the suggested technique outperforms competitor IDSs in terms of prediction accuracy.

In 2020, Bedi et al. [3] proposed a Siam-IDS, which was a novel IDS based on the Siamese Neural Network (Siamese-NN). Without utilizing standard class balancings approaches like oversampling and random under sampling, the recommended Siam-IDS could recognize R2L and U2R assaults. The performance of Siam-IDS was compared to the current IDSs created using DNN and CNN methods. When compared to its competitors, Siam-IDS was able to obtain greater recall values for both R2L and U2R attack classes.

In 2021, Gupta et al. [4] implemented LIO-IDS based on an LSTM classifier and Improved the one-on-one technique for handling both frequent and infrequent network intrusions. LIO-IDS, a two-layer ANIDS had recognized various network intrusions with high precision as well as minimal computational time. Using the LSTM classifier, Layer 1 of LIO-IDS distinguishes intrusions from regular network data. Layer 2 classifies the detected incursions into attack classes using ensemble methods. At the second layer of the suggested LIO-IDS, this study offers an I-OVO approach for performing multi-class classification. Unlike the standard OVO method, the suggested I-OVO method tests each sample with only three classifiers greatly decreasing the testing time. Further, to improve the detection capabilities of the proposed LIO-IDS, oversampling techniques were utilized at Layer 2. For the NSL-KDD, CIDDS-001, and CICIDS2017 datasets, the proposed system's performance was assessed in terms of Accuracy, Recall, Precision, F1-score, ROC curve, AUC values, training time, and testing time.

In 2021, Huang et al. [5] used IGAN to tackle the class imbalance problem. The major uniqueness of the model was that it supplements the traditional GAN with an unbalanced data filter and convolutional layers, resulting in additional primary indicators for minority classes. Furthermore, employing the samples created by IGAN, an IGAN-based IDS, dubbed IGAN-IDS, has been built to deal with class-imbalanced intrusion detection. IGAN-IDS has been made up of three different modules: feature extraction, IGAN, and DNN. To begin, raw network characteristics were transformed into feature vectors using a FNN. After that, the IGAN creates fresh samples inside the spatial domain. Finally, the final intrusion identification was carried out by the DNN, which would also be made up of convolutional and fully-connected layers.

In 2021, Bedi et al. [6] Improved Siam-IDS (I-SiamIDS), a two-layer ensemble for managing class imbalance problems was presented as an algorithm-level method. I-SiamIDS does not use data-level balancing approaches to identify both majority and minority classes at the algorithmic level. To identify the intrusions, the first layer of I-SiamIDS implements hierarchical filtering of input samples using an ensemble of b-XGBoost, Siamese Neural Network (Siamese-NN), and DNN. These attackers were transmitted to I-SiamIDS (second layer), which uses a multi-class eXtreme Gradient Boosting classifier for categorizing attacks into distinct types. For both the NSL-KDD and CIDDS-001 datasets, I-SiamIDS outperformed its competitors in Accuracy, Recall, Precision, F1-score, and values of Area AUC.

In 2021, Lee et al. [7] have solved the data imbalance problem by the GAN model. It has also suggested an RF-based model to identify the classification accuracy after correcting data imbalances with
Despite utilizing standard class balancing approaches like oversampling and random undersampling, Neural Networks was introduced, to address the problem of class imbalance in IDSs (Siamese-NN). The R2L and U2R attack classes have a small sample size. Siames-IDS, a new IDS based on Siamese Denial of Service and Probe attacks in these two datasets involve a large number of cases, while trained on intrusion detection datasets such as KDD and NSL-KDD. Besides the Normal class, the breaches. To resist new and complex attacks, modern IDSs have been created using DL techniques and algorithms to identify intrusions. Over the last few decades, "ML and DM" techniques have been generally used in IDS for identifying intrusions. Improving intrusion categorization accuracy while lowering the false-positive rate is one of the key problems for creating IDS using ML and data mining algorithms. RNN through consecutive information in an arranged time period as a mean traffic report for the development of the IDS. Then they were tested on three different benchmark datasets with different architecture and parameters using learning rates in both balanced and imbalanced training scenarios, and a supervised-learning method label was provided. Thus, results showed that a single CNN model combined with LSTM outperformed other approaches because it could better identify patterns in network traffic record data.

In 2022, Bo Cao et al. [41] executed a repeated edited nearest neighbors (RENN) and hybrid sampling algorithm combining adaptive synthetic sampling (ADASYN) to solve the inequity in the dataset. Initially, the RF algorithm was used for selection, and Pearson correlation analysis was used to reduce feature redundancy. CNN was used to extract spatial features, which were then processed using attention mechanisms and average pooling and max pooling techniques to give each feature a different weight. This reduced overhead while also enhancing performance. The long-distance-dependent information was gathered through the (GRU) through Softmax. Finally, this system was tested with various datasets to improve the accuracy rate compared to other CNN-GRUs while dealing with class imbalance problems.

In 2022, Azizjon Meliboev et al. [42] experimented with DL for instance, GRU, CNN, LSTM, and RNN through consecutive information in an arranged time period as a mean traffic report for the development of the IDS. Then they were tested on three different benchmark datasets with different architecture and parameters using learning rates in both balanced and imbalanced training scenarios, and a supervised-learning method label was provided. Thus, results showed that a single CNN model combined with LSTM outperformed other approaches because it could better identify patterns in network traffic record data.

In 2022, Fu et al. [43] have executed a DLNID for traffic anomaly detection. A Bi-LSTM network joined with DLNID obtained a sequence feature through a CNN network. ADASYN was used to avoid an imbalance in sample expansion of minority class samples to form a relatively symmetric dataset. To improve information fusion, data dimensionality was also reduced using a modified stacked auto encoder. The manual feature extraction technique is not necessary for DLNID. Experimental results on the publicly available benchmark dataset for network intrusion detection (NSL-KDD) demonstrate that this model's accuracy and F1 score were superior to those of other comparable techniques. For publicly available benchmark datasets, this strategy has been successfully tested.

2.2 Problem Statement

In the current digital era, network intrusion categorization has become a large and essential challenge in the ICT industry due to the unbalanced big data environment. IDSs are a popular tool for detecting and preventing internal and external network attacks/intrusions at the moment. Misuse-based IDSs generally are split into "Host-based and network-based systems", and employ pattern-matching algorithms to identify intrusions. Over the last few decades, "ML and DM" techniques have been frequently used in IDS for identifying intrusions. Improving intrusion categorization accuracy while lowering the false-positive rate is one of the key problems for creating IDS using ML and data mining techniques. R2L and U2R cyber attacks are some of the datasets' minority classes, and DL-based IDSs seem unable to identify them adequately owing to a paucity of sample points in these classes. This creates an issue of class imbalance and raises the risk of the network being hacked through unnoticed breaches. To resist new and complex attacks, modern IDSs have been created using DL techniques and trained on intrusion detection datasets such as KDD and NSL-KDD. Besides the Normal class, the Denial of Service and Probe attack classes in these two datasets involve a large number of cases, while the R2L and U2R assault classes have a small sample size. Siames-IDS, a new IDS based on Siamese Neural Networks was introduced, to address the problem of class imbalance in IDSs (Siamese-NN). Despite utilizing standard class balancings approaches like oversampling and random undersampling,
the suggested Siam-IDS effectively identifies R2L and U2R intrusions. Siam-IDS, a new IDS based on Siamese Neural Networks was introduced, to address the problem of class imbalance in IDSs (Siamese-NN). Despite utilizing standard class balancing approaches like oversampling and random under sampling, the suggested Siam-IDS effectively identifies R2L and U2R intrusions. On the benchmark Cybersecurity datasets KDD99, UNSW-NB15, UNSW-NB17, and UNSW-NB18, the resampling methods random under sampling, random oversampling, random under sampling, and random oversampling, random under sampling with Synthetic Minority Oversampling Technique, and random under sampling with Adaptive Synthetic Sampling Method were used. The findings were assessed using "Macro precision, Macro recall, and Macro F1-score". The following trends were discovered: first, oversampling increases training time while under sampling lowers it; second, if the data is very unbalanced, both oversampling and under sampling greatly enhance recall. third, resampling will not have much of an influence if indeed the information is not highly unbalanced; fourth, with resampling, primarily oversampling, more of the minority data (attacks) was discovered.

3. Proposed IDS in Imbalanced Dataset

3.1 Architectural Description

The following four key phases are used to establish an intrusion detection model for class imbalance data: (a) Pre-processing, (b) Imbalance processing, (c) Feature extraction, and (d) Intrusion detection phase. Fig. 1 depicts the general architecture of the planned project. The obtained data $D^o$ on class imbalance is first pre-processed using a data cleaning and standardization approach. Then, the pre-processed class imbalance data $D^o$ is more balanced in the imbalanced processing phase by an improved over-sampling technique. The imbalanced processing phase uses a new enhanced over-sampling approach using SMOTE and Multi-kernel FCM clustering model to balance the pre-processed class imbalance data. Subsequently, multi-features like EMA, DEMA, IWH, and statistical features (Mean, Median, Standard Deviation, Percentile, Momentum, Min-max, Skewness, and Kurtosis) are extracted from these balanced data. The extracted overall features $F$ are given for K-fold validation. The data feature acquired from the k-fold model denoted as $F^*$ is used to train the classifier in the intrusion detection phase. Then, the presence of an intruder is detected by a combined ML model, which is the combination of four individual ML models such as SVM ($Out_{SVM}$), RF ($Out_{RF}$), optimized DBN ($Out_{DBN}$) with incremental learning, and NB ($Out_{NB}$) to achieve the utmost detection performance. Moreover, to achieve the utmost detection accuracy as well as better tradeoff performance, we’ve fine-tuned the weight of DBN via a new SI-SOA, which is an improved version of standard SOA. Then, the logoff performance is computed from the outcome acquired from each of the individual classifiers (SVM, DBN, NB, and RF). Finally, the log computed result will portray the detected attacks in the network.
Preprocessing the data is an important mining approach that transforms raw data into a usable and accessible structure. Data preprocessing is a collection of approaches being used applying a data mining approach, and this is regarded to be of greatest concern inside the well-known Knowledge Discovery from Data process. In this work, the collected raw data is pre-processed via data cleaning and data standardization approach.

4. Pre-processing via Data Cleaning and Data Standardization

Preprocessing the data is an important mining approach that transforms raw data into a usable and accessible structure. Data preprocessing is a collection of approaches being used applying a data mining approach, and this is regarded to be of greatest concern inside the well-known Knowledge Discovery from Data process. In this work, the collected raw data is pre-processed via data cleaning and data standardization approach.
4.1 Data Cleaning

The data cleaning/cleansing is used to remove the redundant and meaningless data available in $D^w$ [1]. The data acquired at the end of the data cleaning phase is denoted as $D^{clean}$, which is subjected to the data standardization phase.

4.2 Data Standardization

It is done to scale the data values in a specified range. In this research work, the cleaned data is standardized $D^{Norm}$ using Eq. (1).

$$D^{Norm} = \frac{D^{clean} - \mu}{\delta}$$

Here, $\mu$ and $\delta$ denotes the mean and standard deviation. The data is normalized to a Gaussian distribution with $\mu = 0$ and $\delta = 1$. The data normalization aids in improving the convergence speed and the accuracy of the model. The acquired pre-processed data is denoted as $D^{Norm}$. The final data set generated by a successful data preparation stage is considered a credible and acceptable source for any data mining method performed later. Then, the pre-processed class imbalance data $D^{Norm}$ is more balanced in the imbalanced processing phase by an improved over-sampling technique.

5. Imbalance Processing: an improved over-sampling technique with SMOTE and Multi-kernel FCM clustering model

Inherently, the number of aberrant samples $D^{Norm}$ is low. However, relying only on over-sampling might result in an excessive amount of duplicate data, as well as increased time and space expenditures. To address this, a new enhanced oversampling approach based on SMOTE is developed, with a Multi-kernel FCM clustering model. The current research work oversamples classes that are below $D^{resample}$ to $D^{balanced}$ using SMOTE.

5.1 SMOTE

SMOTE [1] is a traditional oversampling approach that “Synthesizes” minority class samples to expand the amount of minority class samples. To avoid over fitting throughout the phase of constructing a classification model, the "synthesis" is used to generate a sample that does not exist in the original dataset, rather than simply copying samples like ROS. SMOTE creates a new synthetic sample by randomly selecting a location between the minority class sample and its k-nearest neighbors. We utilise a Multi-kernel FCM clustering model to under-sample the majority of class samples to $D^{resample}$ for classes with more samples than $D^{resample}$.

5.2 Multi-Kernel FCM Clustering

One of the most promising fuzzy clustering algorithms is FCM. It's much more versatile than that of the analogous hard clustering methods in most instances. The kernel has been added to the FCM algorithm, which leads to improved performance. The MKFC technique chooses the appropriate membership values in the very same timeframe. Effective kernels or attributes are more attributable to clustering and thereby enhance the outcomes. Ultimately, MKFC produces fuzzy (soft) prediction performance, which seems to be best suited to situations wherein clusters overlap significantly. In this research work, 3 major kernels are used: RBF, sigmoid, and cosine.

1. **Cosine kernel:** The L2-normalized dot product of vectors is computed using cosine similarity. If $x, y$ are row vectors, the cosine similarity $\kappa_{n_{\text{cos}}}$ between them is defined in Eq. (2).

$$\kappa_{n_{\text{cos}}}(x, y) = \frac{x^T}{\|x\| \|y\|}$$

Because Euclidean (L2) normalization projects the coordinates onto the unit sphere, their dot product equals the cosine of the angle between the locations represented by the vectors. This is known as cosine similarity.

2. **Sigmoid kernel:** The sigmoid kernel between two vectors is computed using the function sigmoid kernel. The hyperbolic tangent, or Multilayer Perceptron, is another name for the sigmoid kernel $\kappa_{n_{\text{sig}}}$ (because, in the neural network field, it is often used as a neuron activation function). It is defined as in Eq. (3)
A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

\[
\text{ker}_{\text{sig}}(x, y) = \tanh(\gamma x^T y + c0)
\]  

(3)

where: \(x, y\) are the input vectors; \(\gamma\) refers to the slope, \(c0\) stands for intercept.

3. **RBF kernel**: The RBF kernel \(\text{ker}_{\text{RBF}}\) between two vectors typically is computed by the function RBF kernel. This kernel's definition is as in Eq. (4).

\[
\text{ker}_{\text{RBF}}(x, y) = \exp(-\|x - y\|^2)
\]  

(4)

The input vectors \(x, y\) are used. The Gaussian kernel with variance \(\sigma^2\) is known as \(\gamma = \sigma - 2\). From the balanced dataset, the multi-features are extracted. This balanced dataset is denoted as \(D^{\text{balanced}}\).

**Algorithm 1**: Pseudo-code of new improved over-sampling technique with SMOTE and Multi-kernel FCM clustering model

**Input**: \(D^{\text{Norm}}_i = \{D^{\text{Norm}}_1, D^{\text{Norm}}_2, ..., D^{\text{Norm}}_M\}\)

Here, \(M\) denotes the overall count of classes

The overall count of samples is pointed as \(|D^{\text{Norm}}_i| = N\)

**Output**: Balanced dataset \(D^{\text{balanced}}\)

Compute \(D^{\text{sample}} = \text{int}\left(\frac{N}{M}\right)\)

for \(i \leftarrow 1\) to \(M\) do

\[
\text{if } |D^{\text{Norm}}_i| < D^{\text{sample}} \text{ then}
D^{\text{balanced}}_i = \text{SMOTE}(D^{\text{Norm}}_i, D^{\text{sample}}) \# \text{D}^{\text{balanced}}_i \text{ is oversampled using SMOTE, so } D^{\text{Norm}}_i = D^{\text{balanced}}_i
\]

endif

if \(|D^{\text{Norm}}_i| < D^{\text{sample}}\) then

\[G_k = MK - \text{FCM}(D^{\text{Norm}}_i, M)\] \# to cluster \(D^{\text{Norm}}_i\) into \(M\) clusters, the Multi-kernel FCM MK - FCM clustering model is used

For \(k \leftarrow 1\) to \(M\) do

\[G^i_k = \text{Sample}\left(G_k, \frac{D^{\text{sample}}}{M}\right)\]

\#select \(\frac{D^{\text{sample}}}{M}\) randomly from \(G_k\)

end for

\[D^{\text{balanced}}_i = \text{Concatenate}\left(G^i_k\right)\]

endif

\[D^{\text{balanced}} = \text{Concatenate}\left(D^{\text{balanced}}_i\right)\]

end for

return \(D^{\text{balanced}}\)

6. **Multi-Feature Extraction**

The multi-features like EMA, DEMA, IWH, and statistical features (Mean, Median, Standard Deviation, Percentile, Standardized moment) are extracted from these balanced data \(D^{\text{balanced}}\).

6.1 **Statistical Features**

The statistical features like Mean, Median, Min and Max, Skewness, and Variance are extracted from \(D^{\text{balanced}}\).

**Mean**: Within the time window, \(\mu\) is the average of \(R\) values of \(D^{\text{balanced}}_i = \{D^{\text{balanced}}_1, D^{\text{balanced}}_2, ..., D^{\text{balanced}}_R\}\). This is mathematically given in Eq. (5).
A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

\[
\mu = \frac{1}{R} \sum_{i=1}^{R} D_i^{\text{balanced}}
\]

Standard Deviation \((\sigma)\): It is computed “To measure how the data values \(D_i^{\text{balanced}} = \{D_1^{\text{balanced}}, D_2^{\text{balanced}}, ..., D_R^{\text{balanced}}\}\) are spread out”.

Median: When a data set is ordered, the value that falls in the center of the data. “The median is the middle data entry in a data collection with an odd number of items. If there are an even number of items in the data, the median is calculated by adding the two values in the center and dividing the result by two.”

Min and Max: “The maxima and minima (The plurals of maximum and minimum) of a function, known collectively as Extrema (Plural of extremum), are the largest and smallest value of the function, either within a given range (The local or relative extrema) or on the entire domain of a function (the global or absolute extrema)”, according to mathematical analysis.

Skewness [36] \(f^{\text{skewness}}\): “It’s a symmetry measure, or more precisely, the lack of symmetry. Only if the left and right sides of the center point are comparable is a dataset or distribution symmetric.” The mathematical expression of skewness \(S\) is given in Eq. (6).

\[
f^{\text{skewness}} = \frac{\sum_{i=1}^{M} (D_i^{\text{balanced}} - \mu)^3}{\sigma^3} / R
\]

For any symmetric data, the skewness value is near 0, and for the normal distribution, it is zero.

Kurtosis [37] \(f^{\text{kurtosis}}\): “It’s a statistic that determines whether data is light-tailed or heavy-tailed, and it’s connected to the normal distribution”. Mathematical formula of kurtosis \(Z\) for univariate data such as \(D_i^{\text{balanced}} = \{D_1^{\text{balanced}}, D_2^{\text{balanced}}, ..., D_R^{\text{balanced}}\}\) is expressed in Eq. (7).

\[
f^{\text{kurtosis}} = \frac{\sum_{i=1}^{M} (D_i^{\text{balanced}} - \mu)^4}{\sigma^4} / R
\]

Standardized Moment: In probability theory and statistics, a standardized moment of a probability distribution is a normalized moment (usually a higher degree central moment). The normalization appears to be a standard deviation term, which renders the moment scale invariant. The standardized moment is calculated mathematically using Eq. (8)

\[
\mu_m = E\left[\left(D_i^{\text{balanced}} - \mu\right)^m\right] = \int_{-\infty}^{\infty} \left(D_i^{\text{balanced}} - \mu\right)^m \cdot B(D_i^{\text{balanced}})
\]

Here, probability distribution and expected value \(D_i^{\text{balanced}}\) are pointed as \(B\) and \(E\) respectively. For the degree \(m\), the standardized moment is denoted as \(\mu_m\), which is said to be the \(m^{th}\) moment of the mean. In addition, \(\sigma_m\) denotes the \(m^{th}\) power of standard deviation.

Percentile: It provides the idea of “How the data values \(D_i^{\text{balanced}}\) are spread over the interval from the smallest value to the largest value”. The extracted statistical features are denoted as \(F^{\text{statistical}}\).

A. EMA

An EMA [38] is indeed a kind of MA that gives the much more recent data points more weight and importance. The exponentially weighted MA is yet another name for the exponential MA. The extracted EMA features are denoted as \(F^{\text{EMA}}\). This is mathematically given in Eq. (9).

\[
F^{\text{EMA}} = \frac{1}{R} \sum_{i=1}^{R} D_i^{\text{balanced}}
\]

The derived EMA features are indicated by \(F^{\text{EMA}}\).

B. Proposed weighted Holoentropy features

The holo-entropy \(HL_{\nu}(\nu)\) is defined as the sum of the entropy and the total correlation of the random vector \(\nu\), and can be expressed by the sum of the entropies on all attributes. It is modeled as in Eq. (10). The weighted holo-entropy \(W_{\nu}(\nu)\) is the sum of the weighted entropy on each attribute of the random vector \(\nu\)” and it is modeled as in Eq. (11).

\[
HL_{\nu}(\nu) = H_{\nu}(\nu) + \sum_{\nu} H_{\nu}(D_i^{\text{balanced}})
\]

\[
W_{\nu}(\nu) = \sum_{\nu} H_{\nu}(D_i^{\text{balanced}})
\]
The weight function $w_{i}^{(\phi)}(D_{i}^{\text{balanced}})$ is computed using the newly developed mathematical expression, which is based on the tanh activation function given in Eq. (12).

$$w_{i}^{(\phi)}(D_{i}^{\text{balanced}}) = \left( \frac{2}{1 + \exp(2(-H_{i}(D_{i}^{\text{balanced}})))} \right)^{-1}$$

The extracted proposed weighted holoentropy feature is denoted as $F^{\text{IWH}}$.

C. DEMA

The DEMA is a mixture of smoothed EMA and a standard EMA. The extracted DEMA feature is denoted as $F^{\text{DEMA}}$. The extracted overall feature is denoted as $F = F^{\text{IWH}} + F^{\text{statistical}} + F^{\text{EMA}} + F^{\text{DEMA}}$, which is given for K-fold validation.

7. K-Fold Validation

Cross-validation is indeed a re-sampling method for evaluating ML algorithms on a small set of data. The technique includes only one parameter, k, which specifies how many groups a given data sample should be divided into. As a result, k-fold cross-validation is a popular name for the process.

The general procedure is as follows:

1. Randomly shuffle the dataset feature $F$.
2. Divide $F$ into a total of $k$ groups.
   For each one-of-a-kind group:
   a) Use the group as a test data set or a holdover.
   b) As a training data set, use the remaining groupings.
   c) Create a model and test it on the training set.
   d) Keep the average rating but discard the model out.
   e) Using a sampling of model assessment ratings, summarise the model’s ability.

The data feature acquired from the $K$-fold model denoted as $F^{*}$ is used to train the classifier in the intrusion detection phase. In the result and discussion phase, this $K$ value is varied, and the performance of the projected model is recorded.

8. Intrusion Detection Phase

The intrusion detection phase is modeled with four individual ML models like SVM, Optimized DBN with incremental learning, NB, and RF to achieve the utmost detection performance. Each of the classifiers is a trained acquired data feature $F^{*}$. The final result is the logoff of outcomes from each of the classifiers.

8.1 RF based Classification

RF is indeed a supervised classification approach that is based on the ML model's DT [36]. The RF, in general, is the combination of "unbiased, unconnected, and de-correlated DT," thus the name “RF”. The decision tree technique is used to create the tree-like model, and each tree is created by randomly picking nodes. The significant concepts used to determine the results are entropy and Gini values. The Gini coefficient is used to calculate impurity, and the Entropy is used to calculate the information gain of nodes. The node with the least Gini impurity is selected for further decomposition during the computation of the Gini impurity. Moreover, from the Gini value, the impurity can be calculated by subtracting the square of the probability $\text{prob}$ of each class from 1. Furthermore, by subtracting the square of each class's probability from 1, the impurity can be calculated from the Gini value. Eq. (13) is used to calculate it mathematically. It is said to be a perfect split when $Gini = 0$. The time is indicated as $t$.

$$Gini = 1 - \sum_{i=1}^{c} \text{prob}_i^2$$  

(13)

Entropy is also used to divide the node with the most information gain. When $\text{Entropy gain} = 1$ at the leaf node, it is considered to be a full information-acquired node and hence a perfect split. The entropy ($\text{Entropy}$) is computed using the formula Eq. (14).

$$\text{Entropy} = \sum_{i=1}^{c} -\text{prob}_i \log_2 \text{prob}_i$$

(14)
The information gain is determined using the computed entropy value and the difference between the computed entropy and the other classes. As a result, RF is regarded as one of the finest supervised classification algorithms, providing improved classification while avoiding overfitting. The outcome from RF is denoted as $Out_{RF}$.

8.2 Naive Bayers

It's a categorization method that uses Bayes' Hypothesis and assumes predictor independence. An NB classifier assumes that one feature in a class is independent of the other attribute. The NB model is very simple to construct and is especially effective when dealing with big datasets. NB is renowned for surpassing only the most advanced classification algorithms due to its simplicity. The Bayes theorem allows us to calculate posterior probability $P(C|F^*)$ from $P(C), P(F^*)$. Consider the following equation in Eq. (15)

$$P(C|F^*) = \frac{P(C|F^*)P(C)}{P(F^*)}$$

(15)

Here, $P(C|F^*)$ is the likelihood and $P$ denotes the posterior probability. In addition, $P(C)$ and $P(F^*)$ denotes the class prior probability and predictor prior probability. The outcome from Naive Bayes is denoted as $Out_{NB}$.

8.3 SVM

SVM [39] is well-known for its straightforward non-linear regression procedure, which also makes it easy to solve the optimization problem:

$$-L(\lambda) = -\sum_{d=1}^{Q} \lambda_d +$$

$$\frac{1}{2} \sum_{d=1}^{Q} \sum_{i=1}^{d} \lambda_d \lambda_i y_d y_i \lambda(z_d, z_i) \rightarrow \min_{\lambda}$$

(16)

Here, $z_d$ and $\lambda(z_d, z_i)$ denotes the elements in training data features $F^*$ and kernel function respectively. In addition, $\chi_d$ and $y_d$ represents the dual variable and the number that is +1 or -1. Further, $Cr$ is the regularization parameter, $Q$ denotes the count of objects that are said to be available $F^*$. To enhance the precision of the SVM order, it's important to narrow down the region of the improperly grouped items. The isolating hyperplane is where the majority of the improperly placed items are found. As a result, using the additional tools to improve the order quality of the items within the isolating strip is critical. The outcome from SVM is denoted as $Out_{SVM}$.

8.4 Optimized DBN with Incremental Learning

Smolensky introduced DBN [40] in 1986, and it consists of three layers: "Input, Output, and Hidden". The visible neurons are in the input layer, whereas the hidden neurons are in the output layer. There is an association between hidden neurons and input neurons. There is a unique symmetrical link between the visible and hidden neurons. The stochastic neuron model may be used to extract the relevant output from DBN for each input. In addition, the Boltzman network is used in DBN to achieve probabilistic results. The DBN result is represented by the notation $r$ in binary form. The probability $J_p(\delta)$ in the sinusoidal function is expressed in Eqs. (17) and (18). When $S > 0$, the pseudo temperature parameter $S$ reduces the probability noise level. Furthermore, Eq. (19) shows the stochastic probability model in a deterministic form.

$$r = \begin{cases} 1 & \text{with } J_p(\delta) \\ 0 & \text{with } 1 - J_p(\delta) \end{cases}$$

(17)

$$J_p(\delta) = \frac{1}{1 + e^{\frac{\delta}{S}}}$$

(18)

$$\lim_{S \to 0^+} J_p(\delta) = \lim_{S \to 0^+} \frac{1}{1 + e^{\frac{\delta}{S}}} = \begin{cases} 0 & \text{for } \delta < 0 \\ \frac{1}{2} & \text{for } \delta = 0 \\ 1 & \text{for } \delta > 0 \end{cases}$$

(19)
The energy factor is important for designing the neuron states $h$ of the Boltzmann machine, and the mathematical formula for the energy of the Boltzmann machine is Eq. (20). Furthermore, in DBN, the weight between the neuron and the biases of the neurons is represented by $M_{ji}$ and $\alpha$ respectively. The joint configuration between the visible $q$ as well as hidden neurons $d$ in terms of energy is indicated in Eq. (21), Eq. (22), Eq. (23), and Eq. (24). The binary states of the visible state and the hidden state $i$ and $j$ are represented as $h_i$ and $h_j$ respectively.

\[
E_g(h) = -\sum_{i,j} L_{ij} h_i h_j - \sum_i \alpha_i h_i
\]

\[
\Delta E_g(h_i) = \sum_j L_{ij} h_j + \alpha_i
\]

\[
E_g(\vec{q}, \vec{d}) = -\sum_{(i,j)} L_{ij} q_i d_j - \sum_i q_i u_i - \sum_j d_j g_j
\]

\[
\Delta E_g(q_i, d_j) = \sum_j L_{ij} d_j + u_i
\]

\[
\Delta E_g(q_i d_j) = \sum_j L_{ij} q_i + g_j
\]

The weight parameters derived from the encoded probability distribution of the input data are used to determine the RBM learning pattern. Eq. (25) determines RBM’s weighting assignment, and the assigned probability can be maximized by RBM. $q$ stands for the input visible vector. RBM also can assign probability to each unique visible and hidden vector, as illustrated by the energy function in Eq. (26). The allocated weight is denoted by the notation $G_e$, while the training set is denoted by the symbol $O$. The partition function $V$ is then derived by summing the energy of all possible states of the neurons, as shown in Eq. (27).

\[
G_e = \max_G \prod_{q \in O} P(\vec{q})
\]

\[
P(\vec{q}, \vec{d}) = \frac{1}{V} e^{-E_g(\vec{q}, \vec{d})}
\]

\[
V = \sum_{\vec{q}, \vec{d}} e^{-E_g(\vec{q}, \vec{d})}
\]

Furthermore, the normal Boltzmann Machine does not rely on the visible or hidden neurons during the assessment of energy between the visible or hidden neurons, but the RBM does. RBM is good at data reconstruction but not so good at data categorization; therefore unsupervised learning is used to train RBM. Because RBM takes a long time to converge with the required model, a CD is used. The hidden neurons are used as input in the training patterns MLP layer, which is identical to the RBM layer.

The error function “MSE” is computed in DBN to know about the error interrupted within it during the training process. Mathematically, the MSE is given in Eq. (28).

\[
MSE = Act - Pre
\]

Here, $Act$ points to the actual output (i.e. actual outcome) and $Pre$ denotes the predicted output (i.e. predicted outcome acquired with proposed work). This error ought to be lower, to prove that the projected prediction model is more accurate in detecting the attacks during the testing phase. We’ve attempted to lessen this error (MSE) by using the new SI-SOA model. The objective function or fitness function of this work is shown in Eq. (29).

\[
Obj = \min(MSE)
\]

To achieve this objective, the weight $L$ of DBN is fine-tuned using the SI-SOA model. The solution fed as input to SI-SMO is $L$. Here, $p$ denotes the overall weight of DBN. The solution encoding model is shown in Fig.2.

![Fig.2. Solution Encoding](image)
A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

Below is a list of the stages involved in the CD algorithm.

1. The binary input is assumed by picking the training samples \( q \) and attaching them to the visible neurons.

2. The probability of the hidden neurons \( U_d \) is calculated by multiplying the visible vector \( q \) with the weight matrix \( G \), which is denoted by \( U_d = \psi(Gq) \) in the Eq. (30).

\[
p(d_j \rightarrow 1|\hat{q}) = \psi\left( s_j + \sum q_iL_{i,j}\right)
\]  

(30)

3. The hidden states \( d \) are sampled using probability \( U_d \).

4. The positive gradient \( \zeta^{+} \) is calculated as \( \zeta^{+} = qU_d^T \) as the outer product of \( q \) and \( U_d \).

5. According to Eq. (31), the visible state \( q^{'} \) is reconstructed from the hidden state \( \hat{d} \). Furthermore, the visible state \( q^{'} \) is reconstructed by resampling the concealed states \( \hat{d} \).

\[
p(q_i \rightarrow 1|\bar{d}) = \psi\left( u_i + \sum_j d_jL_{i,j}\right)
\]  

(31)

6. The exterior product of vectors \( q \) and \( \hat{d} \) is the negative gradient \( \zeta^{-} \). \( \zeta^{-} = q\hat{d}^T \) is the formula for calculating the negative gradient.

7. The weight updates are obtained by subtracting the negative gradient \( \zeta^{-} \) from the positive gradient \( \zeta^{+} \). This weight update is carried out by Eq. (32)

\[
\Delta G = \chi(\zeta^{+} - \zeta^{-})
\]  

(32)

8. The updating of weights occurs using the newly acquired values, as described in Eq. (33)

\[
\Delta L_{i,j} = \Delta L_{i,j} + L_{i,j}
\]  

(33)

The outcome from DBN is denoted as Out\_{\text{obs}}.

Incremental Learning: In a nutshell, incremental learning is indeed a continual learning process in which information classes of labeled data are progressively made accessible in batches. The phrase "Incremental Learning" is also used in the literature about incremental network pruning and growth, as well as online learning. In addition, terms including lifelong learning, constructive learning, and evolutionary learning have been utilized to determine the acquisition of new information. To imitate genuine, biological brains, a pure incremental learning model must be developed. Animals and humans can remember new events despite forgetting previous ones because of the supremacy of the biological brain. In artificial neural networks, however, precise sequential learning doesn't function completely. The adoption of a fixed architecture and/or a training procedure focused on minimizing an objective function, which results in catastrophic interference. It’s because the objective function's minima for one number of instances may vary from the minima of succeeding sets of instances. As a result, each new training dataset induces the network to forget about the prior ones partially or completely. The stability-plasticity conundrum [15] is the name given to this issue. To address these issues, an incremental learning algorithm is used that satisfies the following requirements:

i. It must be able to expand the network and accommodate new tasks (classes) as they are introduced through new examples.

ii. There should be minimal overhead when training for new tasks (classes).

iii. It shouldn't be necessary to have accessibility to the previously observed instances that were used to train the preexisting classifiers.

iv. It must maintain prior knowledge, avoiding catastrophic forgetting. For illustration: Consider that perhaps a base network is trained with 4 task 0 classes (C1–C4), and after training, all training data for all those four classes is deleted. The network must next handle sample data for task 1 with two classes (C5, C6) while maintaining knowledge of the first four classes.

As a result, network capacity should be upgraded, as well as the network must be efficiently retrained using only the new data of task 1 (of C5 and C6), such that the upgraded structure could categorize both task classes (C1 and C6). This is a task-specific categorization if the tasks are categorized individually. When they are categorized jointly, however, it is referred to as combined classification. We'll concentrate on the task-specific situation while considering the combined categorization. The incremental learning approach is depicted in Fig. 3.
**SI-SOA:** The SOA [29] is based on the seagulls’ assaulting behavior. In general, the SOA excels in solving complicated optimization problems with a faster rate of convergence. Furthermore, it is claimed in the literature that self-adaptivity in the conventional optimization model improves convergence and prevents solutions from becoming stuck in local optima [32] [33] [34] [35]. The proposed SI-SOA model is illustrated below:

1. **Step 1:** Set up the search agent’s motion behaviors \(A, B\), as well as the current iteration \(itr\) as well as the maximum iteration \(Max^{itr}\). Furthermore, the populace \(pop\) of the search agent is set up.

2. **Step 2:** Set the constants for the spiral form definition to 1 and the variable for managing the frequency of using variables to \(f_c=2\).

3. **Step 3:** While \(itr < Max^{itr}\) do
   a) Compute the search agent’s fitness \(Fit\) using Eq. (29).
   b) Proceed to the SOA migration step. During migration, the algorithm replicates how a flock of seagulls migrates from one area to another. At this stage, a seagull must satisfy three requirements: Keeping crashes at bay: To avoid collisions between neighbors, a new search agent position is created using an optimum variable \(A\) (i.e., other seagulls).
   c) Rather than utilizing the standard SOA model, we have developed a novel mathematical formula to determine the optimal variable \(A\). The point of the search agents where no collision occurs is mathematically described as Eq. (34). Eq. (35) shows the newly developed equation for computing the search agent’s movement behavior \(A\).

\[
C = A \times H(itr) \tag{34}
\]

\[
A = A^\alpha + \int_1^{itr} V.f(N; \lambda, t)dir \tag{35}
\]

Wherein,

\[
V.f(N; \lambda, t)dir = \frac{itr}{A} \left( \frac{N}{\lambda} \right)^{\alpha-1} \left( \frac{\lambda}{\lambda} \right)^t e^{-\left( \frac{N}{\lambda} \right)^t} \tag{36}
\]

Here, \(N, \lambda\) are the parameters of the Weibull probability distribution and \(S\) is the span factor.

4. Movement in the direction of the best neighbor: In this paper, a new mathematical model for movement in the direction of the best neighbor is created by adding a new levy flight function \(Levy(\beta)\), a random walk displayed by the search agents while seeking the optimum location to update the solutions.

---

**Fig. 3.** Incremental learning: An illustration
e) The performance enrichment of the projected model is also aided by this factor. After avoiding accidents with their neighbors, the search agents head in the direction of the best neighbor. \( \text{Rand1} \) is a random number that is initialized. This process is mathematically described as Eq. (37).

\[
\overline{M} = C + B \times \text{Levy}(\beta) \times \left( \frac{\text{Fit}(\text{itr})}{H(\text{itr})} \right)
\]

(37)

Here, \( \overline{M} \) denotes the positions of the search agent towards the best fitness \( \text{Fit}(\text{itr}) \), and \( C \) is the location of the search agent that doesn’t collide with the other search agents. Mathematically, \( B \) is modeled as per Eq. (38).

\[
B = 2 \times A^2 \times \text{Rand1}
\]

(38)

a) Compute the search agent's fitness \( \text{Fit} \) using Eq. (29).

b) Finally, the search agent can modify its location to the optimal search agent by maintaining tight contact with them. Eq. (39) is used to calculate the distance between \( \overline{\text{Fit}}(\text{itr}) \) and the search agent.

\[
D = \left( \overline{C} + \overline{M} \right)
\]

(39)

b. After that, go to the "attacking phase" to update the search agents' positions.

c. Step 4: End while
d. Step 5: Return \( \text{Fit} \)
e. Step 6: End

Finally, the log-off performance is computed onto the outcomes acquired from each of the classifiers SVM(\( \text{Out}_{\text{svm}} \)), RF(\( \text{Out}_{\text{rf}} \)), optimized DBN(\( \text{Out}_{\text{dbn}} \)) with incremental learning, and NB (\( \text{Out}_{\text{nb}} \))

9. Result and Discussion

9.1 Simulation procedure

The proposed intrusion detection model with class imbalance data was implemented in PYTHON and the outcomes were verified. Dataset 1 and Dataset 2 are gathered from:[30] and [31], respectively. The performance of the proposed work (MULTI-CLASSIFIER+SI-SOA) is compared over the existing models like SGM-CNN [1], NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA respectively. In this research work, we have fixed the K-value of \( K \)-fold evaluation as 5(\( K =5 \)), and we’ve varied this \( K \) value from 1, 2, 3, 4, and 5, to evaluate the performance of the projected model. Furthermore, the performance was based on the different performance measures including “Accuracy, Sensitivity, Specificity, Precision, F-measure, FDR, FNR, FPR, NPV, and MCC” respectively.

9.2 Overall Performance Analysis

The performance analysis of the proposed scheme is computed to the existing schemes like SGM-CNN[1], NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA respectively in terms of certain metrics like “positive measures (Accuracy, Sensitivity, Specificity, Precision); negative measures (FPR and FNR) and %, other measures (F-measure, NPV, and MCC)”, and it is illustrated in Fig. 4, 5 and 6 respectively. On observing the outcomes, the projected model has attained the best performance (higher positive measures and lower negative measures) for every variation in the \( K \)-values; hence the projected model is said to be best for intrusion detection. All these improvements are owing to the extraction of the most relevant features (IWH features) along with the standard features to train the modeled ensemble classifier. Moreover, the count of the abnormal samples has been handled using the improved oversampling model, and this is also a major reason for enhancement in the performance of the projected model. The accuracy of the projected model is higher for every variation in the \( K \)-values. At \( K =5 \), the proposed work attained the highest accuracy value of 92%, which is 13%, 9.7%, 11.4%, 11.55, 11.6%, 11.6%, and 11.8% better than the accuracy value recorded by the existing models like SGM-CNN, NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA, respectively. In addition, the specificity, sensitivity, and precision of detecting the attack by the proposed work are maximum when computed to the existing scheme for every variation in the \( K \)-value. At \( K =1 \), the precision of the projected model in identifying the intrusion in network is 26.3%, 63.15%, 13.6%, 63.15%, 231.05%, 21.05% and 23% better than the existing approaches like SGM-CN, NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA respectively.
Fig. 4. Performance Analysis of Proposed and Traditional Optimization Model: (a) Accuracy, (b) Precision, (c) Specificity, (d) Sensitivity.

Fig. 5. Performance evaluation of the proposed and extant tactics in terms of (a) FPR and (b) FNR.
A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

On the other hand, negative metrics like FPR and FNR of the adopted model over other existing schemes for varying $K$-values are represented in Fig. 5. On observing the outcomes, the proposed work recorded the least error function in terms of FPR and FNR for every variation the $k$-value. As we’ve handled the unbalanced scenario using the newly proposed over-sampling procedure, we could acquire negligible errors. At $K=1$, the FNR of the proposed work is 66.6%, 85.7%, 50%, 85.7%, 60%, 60%, and 62.3% better than the existing approaches like SGM-CN, NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA, respectively. Therefore, it is proved that the adopted model minimizes the detection errors that lead to precise detection of intrusion in the networks.

Fig. 6 illustrates the analysis of other measures such as NPV, MCC, and F-measure of the adopted model than other traditional models. Under every variation in the $K$-value, the projected work attained the highest NPV, MCC, and F-measure. Moreover, the F-measure of the proposed work at $K=4$ is 26.3%, 15.7%, 13.6%, 7.89%, 21%, 23.5% and 23.5% better than the existing approaches like SGM-CN, NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA, respectively. Therefore, the performance of the presented model has shown its improvement over other traditional approaches.

### 9.3 Evaluation of Proposed work with and without Incremental Learning

Table I and Table II manifest the performance evaluation of the projected work for dataset-1 (CICIDS2018) and dataset-2 in terms of incremental outcomes. The Proposed work with multi-classifier and no incremental learning is evaluated over the proposed work with multi-classifier and with incremental learning based over-sampling for varying k-values. At $K=1$, the accuracy of the Proposed work with a multi-classifier and no incremental learning is 0.43% better than the accuracy of the proposed work with a multi-classifier and with incremental learning-based over-sampling. In addition, the accuracy of the Proposed work with multi-classifier and no incremental learning at $K=2$ is 0.849399815, which is 0.18% better than the accuracy of the proposed work with multi-classifier and with incremental learning-based over-sampling. At $K=4$, the accuracy of the Proposed work with a multi-classifier and no incremental learning is 0.206% better than the accuracy of the proposed work with a multi-classifier and with incremental learning-based over-sampling. For dataset-2 (UNSWNB15), the performance evaluation of the Proposed work with multi-classifier and no incremental learning and
with Proposed work with multi-classifier and with incremental learning is evaluated for varying \( K \)-values, and the outcomes acquired are noted in Table II. At \( K = 2 \), the accuracy of the Proposed work with a multi-classifier and no incremental learning is 0.20% better than the accuracy of the proposed work with a multi-classifier and with incremental learning-based over-sampling. At \( K = 2 \), the precision of the Proposed work with a multi-classifier and no incremental learning is 0.604306, which is better than the precision of the proposed work with a multi-classifier and with incremental learning-based over-sampling. Thus, from the analysis, it is clear that the proposed work with incremental learning performs better than the proposed work without incremental learning.

<table>
<thead>
<tr>
<th>Measures</th>
<th>For ( K = 1 )</th>
<th>For ( K = 2 )</th>
<th>For ( K = 3 )</th>
<th>For ( K = 4 )</th>
<th>For ( K = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.779201102</td>
<td>0.782644628</td>
<td>0.769972452</td>
<td>0.773140496</td>
<td>0.773140496</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.892847222</td>
<td>0.894583333</td>
<td>0.888194444</td>
<td>0.889791676</td>
<td>0.889791676</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.854755309</td>
<td>0.857063712</td>
<td>0.848568769</td>
<td>0.850692521</td>
<td>0.850692521</td>
</tr>
<tr>
<td>Precision</td>
<td>0.785944444</td>
<td>0.789166667</td>
<td>0.776388889</td>
<td>0.77317358</td>
<td>0.77317358</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.782434302</td>
<td>0.785892116</td>
<td>0.77317358</td>
<td>0.77317358</td>
<td>0.77317358</td>
</tr>
<tr>
<td>MCC</td>
<td>0.653975844</td>
<td>0.620627161</td>
<td>0.639582161</td>
<td>0.639582161</td>
<td>0.639582161</td>
</tr>
<tr>
<td>NPV</td>
<td>0.889142462</td>
<td>0.890871369</td>
<td>0.885131397</td>
<td>0.885131397</td>
<td>0.885131397</td>
</tr>
<tr>
<td>FPR</td>
<td>0.107152778</td>
<td>0.105416667</td>
<td>0.111180556</td>
<td>0.111180556</td>
<td>0.111180556</td>
</tr>
<tr>
<td>FNR</td>
<td>0.220798988</td>
<td>0.21735372</td>
<td>0.228787879</td>
<td>0.228787879</td>
<td>0.228787879</td>
</tr>
</tbody>
</table>
### Table 2: Evaluation of the proposed work with and without incremental learning for varying $K$-values: Dataset 2

<table>
<thead>
<tr>
<th>Measures</th>
<th>Proposed multi-classifier (SVM, Optimized DBN, NB, and RF) without incremental learning</th>
<th>Proposed multi-classifier (SVM, Optimized DBN, NB, and RF) without incremental learning with incremental learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $K=1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.591873</td>
<td>0.586226</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.798403</td>
<td>0.795556</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.729178</td>
<td>0.725392</td>
</tr>
<tr>
<td>Precision</td>
<td>0.596806</td>
<td>0.591111</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.594329</td>
<td>0.588658</td>
</tr>
<tr>
<td>MCC</td>
<td>0.484718</td>
<td>0.534811</td>
</tr>
<tr>
<td>NPV</td>
<td>0.79509</td>
<td>0.792254</td>
</tr>
<tr>
<td>FPR</td>
<td>0.201597</td>
<td>0.204444</td>
</tr>
<tr>
<td>FNR</td>
<td>0.408127</td>
<td>0.413774</td>
</tr>
<tr>
<td>For $K=2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.596832</td>
<td>0.599311</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.800903</td>
<td>0.802153</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.732502</td>
<td>0.734164</td>
</tr>
<tr>
<td>Precision</td>
<td>0.601806</td>
<td>0.604306</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.599308</td>
<td>0.601798</td>
</tr>
<tr>
<td>MCC</td>
<td>0.563222</td>
<td>0.517572</td>
</tr>
<tr>
<td>NPV</td>
<td>0.79758</td>
<td>0.798824</td>
</tr>
<tr>
<td>FPR</td>
<td>0.199097</td>
<td>0.197847</td>
</tr>
<tr>
<td>FNR</td>
<td>0.403168</td>
<td>0.400689</td>
</tr>
<tr>
<td>For $K=3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.591736</td>
<td>0.592562</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.798333</td>
<td>0.79875</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.729086</td>
<td>0.72964</td>
</tr>
<tr>
<td>Precision</td>
<td>0.596667</td>
<td>0.5975</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.594191</td>
<td>0.595021</td>
</tr>
<tr>
<td>MCC</td>
<td>0.541734</td>
<td>0.55423</td>
</tr>
<tr>
<td>NPV</td>
<td>0.795021</td>
<td>0.795436</td>
</tr>
<tr>
<td>FPR</td>
<td>0.201667</td>
<td>0.20125</td>
</tr>
<tr>
<td>FNR</td>
<td>0.408264</td>
<td>0.407438</td>
</tr>
<tr>
<td>For $K=4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.58719</td>
<td>0.589394</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.796042</td>
<td>0.797153</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.720039</td>
<td>0.727516</td>
</tr>
<tr>
<td>Precision</td>
<td>0.592083</td>
<td>0.594306</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.589627</td>
<td>0.59184</td>
</tr>
<tr>
<td>MCC</td>
<td>0.521286</td>
<td>0.549793</td>
</tr>
<tr>
<td>NPV</td>
<td>0.792739</td>
<td>0.793845</td>
</tr>
<tr>
<td>FPR</td>
<td>0.203958</td>
<td>0.202847</td>
</tr>
<tr>
<td>FNR</td>
<td>0.41281</td>
<td>0.410606</td>
</tr>
<tr>
<td>For $K=5$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.594077</td>
<td>0.592149</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.799514</td>
<td>0.798542</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.730656</td>
<td>0.729363</td>
</tr>
<tr>
<td>Precision</td>
<td>0.599928</td>
<td>0.597083</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.596542</td>
<td>0.594606</td>
</tr>
<tr>
<td>MCC</td>
<td>0.548653</td>
<td>0.512672</td>
</tr>
<tr>
<td>NPV</td>
<td>0.796196</td>
<td>0.795228</td>
</tr>
<tr>
<td>FPR</td>
<td>0.200486</td>
<td>0.201458</td>
</tr>
<tr>
<td>FNR</td>
<td>0.405923</td>
<td>0.407581</td>
</tr>
</tbody>
</table>

#### 9.4 Performance Analysis of Proposed Work: Existing Feature Set

The performance of the classifiers is verified by training them with an existing feature set (Statistical Features, EMA, DEMA, Existing Holomutropy, Percentile, and Moment) in terms of “Accuracy, Sensitivity, Specificity, Precision, F-measure, FDR, FNR, FPR, NPV, and MCC” respectively for varying $k$-values. This evaluation is undergone for dataset1 (CICIDS2018) and dataset2 (UNSWNB15).

**Analysis of dataset1:** The evaluation of dataset1 (CICIDS2018) for “Accuracy, Sensitivity, Specificity, Precision, F-measure, FDR, FNR, FPR, NPV, and MCC” corresponding to diverse $K$-values is shown in Table III-Table VII. For $K=1$, the accuracy of the projected work trained with existing feature set is 26.4%, 62.7%, 19%, 12.87%, 8.5%, 7.57% and 8.5% better than the existing models trained with existing feature set like SGM-CNN[1], NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-
A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA. In addition, F-measure of the proposed work trained with existing feature set at $k=1$ is 24.6%, 61.8%, 17%, 10.7%, 6.3%, 5.3% and 6.3% better than the existing models trained with existing feature set like SGM-CNN[1], NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA, respectively. For $K=2$, the accuracy of the proposed trained with existing feature set is 0.893481, which is better than the existing models trained with the existing feature set like SGM-CNN [1] = 0.678486, NN=0.576547, CNN=0.761219, RNN=0.816251, MULTI-CLASSIFIER+WOA=0.848476, MULTI-CLASSIFIER+MFO=0.848476, MULTI-CLASSIFIER+SOA=0.851247 trained with existing feature set. Similarly, under all other variations in the $k$-value, the proposed work exhibited superior performance. This is owing to the balanced data utilized for attack detection.

For dataset 2: Similarly, in the case of dataset 2, the proposed work trained existing feature set for varying $K$-values ($K=1$ to 5) is shown in Table VII- Table XII respectively. On observing the outcomes the proposed work had exhibited the higher outcomes for every variation in the $K$-value. The proposed work had attained the maximal accuracy as 0.790269 at $K=1$. 0.793213 at $K=2$, 0.78967 at $K=3$, 0.795395 at $K=4$ and 0.792335 at $K=5$.

**Table 3:** Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 1 corresponding to $K=1$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.619835</td>
<td>0.31405</td>
<td>0.682094</td>
<td>0.733884</td>
<td>0.770661</td>
<td>0.778512</td>
<td>0.770523</td>
<td>0.842311</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.8125</td>
<td>0.658333</td>
<td>0.843889</td>
<td>0.87</td>
<td>0.888542</td>
<td>0.8925</td>
<td>0.88472</td>
<td>0.917995</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.747922</td>
<td>0.542936</td>
<td>0.789658</td>
<td>0.824377</td>
<td>0.824377</td>
<td>0.84903</td>
<td>0.848938</td>
<td>0.896598</td>
</tr>
<tr>
<td>Precision</td>
<td>0.625</td>
<td>0.316667</td>
<td>0.687778</td>
<td>0.74</td>
<td>0.77083</td>
<td>0.785</td>
<td>0.776944</td>
<td>0.814114</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.622407</td>
<td>0.315353</td>
<td>0.684924</td>
<td>0.73929</td>
<td>0.773859</td>
<td>0.781743</td>
<td>0.773721</td>
<td>0.826037</td>
</tr>
<tr>
<td>MCC</td>
<td>0.433231</td>
<td>0.336241</td>
<td>0.527073</td>
<td>0.605136</td>
<td>0.609086</td>
<td>0.600797</td>
<td>0.633659</td>
<td>0.599973</td>
</tr>
<tr>
<td>NPV</td>
<td>0.809129</td>
<td>0.655602</td>
<td>0.840387</td>
<td>0.86639</td>
<td>0.884855</td>
<td>0.88797</td>
<td>0.884786</td>
<td>0.935247</td>
</tr>
<tr>
<td>FPR</td>
<td>0.1875</td>
<td>0.341667</td>
<td>0.156111</td>
<td>0.13</td>
<td>0.111458</td>
<td>0.1075</td>
<td>0.111528</td>
<td>0.082005</td>
</tr>
<tr>
<td>FNR</td>
<td>0.380165</td>
<td>0.68595</td>
<td>0.317906</td>
<td>0.266116</td>
<td>0.229339</td>
<td>0.221488</td>
<td>0.229477</td>
<td>0.157689</td>
</tr>
</tbody>
</table>

**Table 4:** Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 1 corresponding to $K=2$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.516253</td>
<td>0.364187</td>
<td>0.639696</td>
<td>0.721763</td>
<td>0.769835</td>
<td>0.769835</td>
<td>0.773967</td>
<td>0.836999</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.760278</td>
<td>0.683611</td>
<td>0.8225</td>
<td>0.863889</td>
<td>0.888125</td>
<td>0.888125</td>
<td>0.890209</td>
<td>0.915666</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.678486</td>
<td>0.576547</td>
<td>0.761219</td>
<td>0.816251</td>
<td>0.848476</td>
<td>0.848476</td>
<td>0.851247</td>
<td>0.893481</td>
</tr>
<tr>
<td>Precision</td>
<td>0.520556</td>
<td>0.367222</td>
<td>0.645</td>
<td>0.727778</td>
<td>0.77625</td>
<td>0.77625</td>
<td>0.780417</td>
<td>0.8087</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.518396</td>
<td>0.365698</td>
<td>0.642324</td>
<td>0.724758</td>
<td>0.773029</td>
<td>0.773029</td>
<td>0.777178</td>
<td>0.820624</td>
</tr>
<tr>
<td>MCC</td>
<td>0.312657</td>
<td>0.489877</td>
<td>0.463127</td>
<td>0.586866</td>
<td>0.620112</td>
<td>0.63481</td>
<td>0.621894</td>
<td>0.587377</td>
</tr>
<tr>
<td>NPV</td>
<td>0.751723</td>
<td>0.680775</td>
<td>0.819087</td>
<td>0.860304</td>
<td>0.88444</td>
<td>0.88444</td>
<td>0.886515</td>
<td>0.933181</td>
</tr>
<tr>
<td>FPR</td>
<td>0.239722</td>
<td>0.316389</td>
<td>0.175</td>
<td>0.136111</td>
<td>0.11875</td>
<td>0.11875</td>
<td>0.109792</td>
<td>0.084343</td>
</tr>
<tr>
<td>FNR</td>
<td>0.483747</td>
<td>0.63813</td>
<td>0.360331</td>
<td>0.278237</td>
<td>0.230165</td>
<td>0.230165</td>
<td>0.226033</td>
<td>0.163001</td>
</tr>
</tbody>
</table>
Table 5: Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 1 corresponding to $K = 3$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.583471</td>
<td>0.312948</td>
<td>0.666667</td>
<td>0.379614</td>
<td>0.772039</td>
<td>0.772176</td>
<td>0.821088</td>
<td>0.84647</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.794167</td>
<td>0.657778</td>
<td>0.836111</td>
<td>0.691389</td>
<td>0.889236</td>
<td>0.889306</td>
<td>0.914167</td>
<td>0.919765</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.723546</td>
<td>0.542198</td>
<td>0.779317</td>
<td>0.586888</td>
<td>0.849954</td>
<td>0.850046</td>
<td>0.883102</td>
<td>0.898967</td>
</tr>
<tr>
<td>Precision</td>
<td>0.588333</td>
<td>0.315556</td>
<td>0.672222</td>
<td>0.382778</td>
<td>0.778472</td>
<td>0.778611</td>
<td>0.828333</td>
<td>0.81839</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.585892</td>
<td>0.314246</td>
<td>0.669433</td>
<td>0.381189</td>
<td>0.775242</td>
<td>0.77588</td>
<td>0.824896</td>
<td>0.830306</td>
</tr>
<tr>
<td>MCC</td>
<td>0.37842</td>
<td>0.375978</td>
<td>0.50382</td>
<td>0.308193</td>
<td>0.662767</td>
<td>0.590433</td>
<td>0.626091</td>
<td>0.616246</td>
</tr>
<tr>
<td>NPV</td>
<td>0.790871</td>
<td>0.655048</td>
<td>0.832642</td>
<td>0.68852</td>
<td>0.885546</td>
<td>0.885615</td>
<td>0.60373</td>
<td>0.936756</td>
</tr>
<tr>
<td>FPR</td>
<td>0.205833</td>
<td>0.342222</td>
<td>0.163889</td>
<td>0.308611</td>
<td>0.110764</td>
<td>0.110694</td>
<td>0.085833</td>
<td>0.080235</td>
</tr>
<tr>
<td>FNR</td>
<td>0.416529</td>
<td>0.687052</td>
<td>0.333333</td>
<td>0.620386</td>
<td>0.227961</td>
<td>0.22782</td>
<td>0.178512</td>
<td>0.15353</td>
</tr>
</tbody>
</table>

Table 6: Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 1 corresponding to $K = 4$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.723416</td>
<td>0.696419</td>
<td>0.63416</td>
<td>0.570799</td>
<td>0.77686</td>
<td>0.768044</td>
<td>0.769972</td>
<td>0.83513</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.864722</td>
<td>0.851111</td>
<td>0.819722</td>
<td>0.787778</td>
<td>0.891667</td>
<td>0.887276</td>
<td>0.888194</td>
<td>0.914838</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.817359</td>
<td>0.799261</td>
<td>0.757525</td>
<td>0.715051</td>
<td>0.853186</td>
<td>0.847276</td>
<td>0.848569</td>
<td>0.892372</td>
</tr>
<tr>
<td>Precision</td>
<td>0.729444</td>
<td>0.702222</td>
<td>0.639444</td>
<td>0.575556</td>
<td>0.783333</td>
<td>0.774444</td>
<td>0.776389</td>
<td>0.806803</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.726418</td>
<td>0.699308</td>
<td>0.636791</td>
<td>0.573167</td>
<td>0.780083</td>
<td>0.771231</td>
<td>0.773167</td>
<td>0.818725</td>
</tr>
<tr>
<td>MCC</td>
<td>0.589357</td>
<td>0.548665</td>
<td>0.454823</td>
<td>0.35932</td>
<td>0.606099</td>
<td>0.613801</td>
<td>0.606741</td>
<td>0.6053</td>
</tr>
<tr>
<td>NPV</td>
<td>0.86134</td>
<td>0.84758</td>
<td>0.816321</td>
<td>0.784509</td>
<td>0.887967</td>
<td>0.885451</td>
<td>0.888409</td>
<td>0.932435</td>
</tr>
<tr>
<td>FPR</td>
<td>0.135278</td>
<td>0.148889</td>
<td>0.180278</td>
<td>0.212222</td>
<td>0.108333</td>
<td>0.112778</td>
<td>0.111806</td>
<td>0.085162</td>
</tr>
<tr>
<td>FNR</td>
<td>0.276584</td>
<td>0.303581</td>
<td>0.36584</td>
<td>0.429201</td>
<td>0.22314</td>
<td>0.231956</td>
<td>0.230028</td>
<td>0.16487</td>
</tr>
</tbody>
</table>

Table 7: Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 1 corresponding to $K = 5$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.626446</td>
<td>0.334435</td>
<td>0.662259</td>
<td>0.370248</td>
<td>0.773554</td>
<td>0.770523</td>
<td>0.770523</td>
<td>0.841583</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.815833</td>
<td>0.66861</td>
<td>0.833889</td>
<td>0.686667</td>
<td>0.89</td>
<td>0.888472</td>
<td>0.888472</td>
<td>0.917442</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.752355</td>
<td>0.556602</td>
<td>0.776362</td>
<td>0.580609</td>
<td>0.85097</td>
<td>0.848938</td>
<td>0.848938</td>
<td>0.895872</td>
</tr>
<tr>
<td>Precision</td>
<td>0.631667</td>
<td>0.337222</td>
<td>0.667778</td>
<td>0.373333</td>
<td>0.78</td>
<td>0.776944</td>
<td>0.776944</td>
<td>0.813424</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.629046</td>
<td>0.338823</td>
<td>0.665007</td>
<td>0.371784</td>
<td>0.776763</td>
<td>0.773721</td>
<td>0.773721</td>
<td>0.825357</td>
</tr>
<tr>
<td>MCC</td>
<td>0.443196</td>
<td>0.409471</td>
<td>0.497176</td>
<td>0.470823</td>
<td>0.61683</td>
<td>0.598398</td>
<td>0.613765</td>
<td>0.586867</td>
</tr>
<tr>
<td>NPV</td>
<td>0.812448</td>
<td>0.66837</td>
<td>0.830429</td>
<td>0.683817</td>
<td>0.85607</td>
<td>0.884786</td>
<td>0.884786</td>
<td>0.934553</td>
</tr>
<tr>
<td>FPR</td>
<td>0.184167</td>
<td>0.331389</td>
<td>0.166111</td>
<td>0.313333</td>
<td>0.11</td>
<td>0.111528</td>
<td>0.111528</td>
<td>0.082558</td>
</tr>
<tr>
<td>FNR</td>
<td>0.373554</td>
<td>0.665665</td>
<td>0.33741</td>
<td>0.629752</td>
<td>0.226446</td>
<td>0.229477</td>
<td>0.229477</td>
<td>0.158417</td>
</tr>
</tbody>
</table>
### Table 8: Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile and Moment) for dataset 2 corresponding to K = 1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.328926</td>
<td>0.334435</td>
<td>0.402755</td>
<td>0.322865</td>
<td>0.540634</td>
<td>0.542837</td>
<td>0.543526</td>
<td>0.679626</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.665833</td>
<td>0.686111</td>
<td>0.703056</td>
<td>0.662778</td>
<td>0.772569</td>
<td>0.773681</td>
<td>0.774028</td>
<td>0.843012</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.552909</td>
<td>0.556602</td>
<td>0.602401</td>
<td>0.548846</td>
<td>0.694829</td>
<td>0.696307</td>
<td>0.696768</td>
<td>0.790269</td>
</tr>
<tr>
<td>Precision</td>
<td>0.331677</td>
<td>0.337222</td>
<td>0.406111</td>
<td>0.325556</td>
<td>0.545139</td>
<td>0.547361</td>
<td>0.548056</td>
<td>0.665999</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.33029</td>
<td>0.335823</td>
<td>0.404426</td>
<td>0.324205</td>
<td>0.542877</td>
<td>0.545090</td>
<td>0.545781</td>
<td>0.672445</td>
</tr>
<tr>
<td>MCC</td>
<td>0.390638</td>
<td>0.317403</td>
<td>0.309213</td>
<td>0.475815</td>
<td>0.456268</td>
<td>0.449025</td>
<td>0.4273</td>
<td>0.56598</td>
</tr>
<tr>
<td>NPV</td>
<td>0.663071</td>
<td>0.665833</td>
<td>0.700138</td>
<td>0.660028</td>
<td>0.769364</td>
<td>0.770471</td>
<td>0.770816</td>
<td>0.849771</td>
</tr>
<tr>
<td>FPR</td>
<td>0.334167</td>
<td>0.331389</td>
<td>0.296944</td>
<td>0.337222</td>
<td>0.227431</td>
<td>0.226319</td>
<td>0.225972</td>
<td>0.15698</td>
</tr>
<tr>
<td>FNR</td>
<td>0.671074</td>
<td>0.665565</td>
<td>0.597245</td>
<td>0.677135</td>
<td>0.459366</td>
<td>0.457163</td>
<td>0.456474</td>
<td>0.320374</td>
</tr>
</tbody>
</table>

### Table 9: Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 2 corresponding to K = 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.450138</td>
<td>0.304683</td>
<td>0.490358</td>
<td>0.344904</td>
<td>0.547107</td>
<td>0.543113</td>
<td>0.547107</td>
<td>0.685137</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.726944</td>
<td>0.653611</td>
<td>0.747222</td>
<td>0.673889</td>
<td>0.775833</td>
<td>0.773819</td>
<td>0.775833</td>
<td>0.84516</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.634164</td>
<td>0.536657</td>
<td>0.661127</td>
<td>0.56362</td>
<td>0.699169</td>
<td>0.696491</td>
<td>0.699169</td>
<td>0.793213</td>
</tr>
<tr>
<td>Precision</td>
<td>0.453889</td>
<td>0.307222</td>
<td>0.494444</td>
<td>0.347778</td>
<td>0.551667</td>
<td>0.547639</td>
<td>0.551667</td>
<td>0.672722</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.452066</td>
<td>0.305947</td>
<td>0.492393</td>
<td>0.346335</td>
<td>0.549378</td>
<td>0.545367</td>
<td>0.549378</td>
<td>0.678614</td>
</tr>
<tr>
<td>MCC</td>
<td>0.312728</td>
<td>0.329399</td>
<td>0.312948</td>
<td>0.337092</td>
<td>0.453992</td>
<td>0.428943</td>
<td>0.461712</td>
<td>0.575294</td>
</tr>
<tr>
<td>NPV</td>
<td>0.723928</td>
<td>0.650899</td>
<td>0.744122</td>
<td>0.671093</td>
<td>0.772614</td>
<td>0.770609</td>
<td>0.772614</td>
<td>0.847867</td>
</tr>
<tr>
<td>FPR</td>
<td>0.273056</td>
<td>0.345389</td>
<td>0.252778</td>
<td>0.326111</td>
<td>0.224167</td>
<td>0.226181</td>
<td>0.224167</td>
<td>0.15484</td>
</tr>
<tr>
<td>FNR</td>
<td>0.549862</td>
<td>0.695317</td>
<td>0.509642</td>
<td>0.655096</td>
<td>0.452893</td>
<td>0.456887</td>
<td>0.452893</td>
<td>0.314863</td>
</tr>
</tbody>
</table>

### Table 10: Evaluation Of The Proposed Work Trained With existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) For dataset 2 corresponding to K = 3

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.338292</td>
<td>0.434711</td>
<td>0.400551</td>
<td>0.338292</td>
<td>0.543664</td>
<td>0.541873</td>
<td>0.539945</td>
<td>0.678914</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.670556</td>
<td>0.719167</td>
<td>0.701944</td>
<td>0.670556</td>
<td>0.774097</td>
<td>0.773194</td>
<td>0.772222</td>
<td>0.84255</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.559187</td>
<td>0.623823</td>
<td>0.600923</td>
<td>0.559187</td>
<td>0.696861</td>
<td>0.69566</td>
<td>0.694367</td>
<td>0.78967</td>
</tr>
<tr>
<td>Precision</td>
<td>0.341111</td>
<td>0.438333</td>
<td>0.403889</td>
<td>0.341111</td>
<td>0.548194</td>
<td>0.546389</td>
<td>0.544444</td>
<td>0.665461</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.339696</td>
<td>0.436515</td>
<td>0.402213</td>
<td>0.339696</td>
<td>0.54592</td>
<td>0.544122</td>
<td>0.542185</td>
<td>0.671829</td>
</tr>
<tr>
<td>MCC</td>
<td>0.469637</td>
<td>0.380566</td>
<td>0.36512</td>
<td>0.459014</td>
<td>0.433159</td>
<td>0.435758</td>
<td>0.439827</td>
<td>0.53911</td>
</tr>
<tr>
<td>NPV</td>
<td>0.667773</td>
<td>0.716183</td>
<td>0.699032</td>
<td>0.667773</td>
<td>0.770885</td>
<td>0.769986</td>
<td>0.769018</td>
<td>0.845474</td>
</tr>
<tr>
<td>FPR</td>
<td>0.329444</td>
<td>0.280833</td>
<td>0.298056</td>
<td>0.329444</td>
<td>0.225903</td>
<td>0.226806</td>
<td>0.227778</td>
<td>0.15745</td>
</tr>
<tr>
<td>FNR</td>
<td>0.661708</td>
<td>0.565289</td>
<td>0.599449</td>
<td>0.661708</td>
<td>0.456336</td>
<td>0.458127</td>
<td>0.460055</td>
<td>0.321086</td>
</tr>
</tbody>
</table>
Table 11: Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 2 corresponding TO $K=4$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.364187</td>
<td>0.323967</td>
<td>0.500275</td>
<td>0.372452</td>
<td>0.548209</td>
<td>0.553444</td>
<td>0.547245</td>
<td>0.687333</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.683611</td>
<td>0.663333</td>
<td>0.752222</td>
<td>0.687778</td>
<td>0.776389</td>
<td>0.779028</td>
<td>0.775903</td>
<td>0.846863</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.576547</td>
<td>0.549584</td>
<td>0.520872</td>
<td>0.699906</td>
<td>0.703416</td>
<td>0.699261</td>
<td>0.795395</td>
<td>0.846863</td>
</tr>
<tr>
<td>Precision</td>
<td>0.367222</td>
<td>0.326667</td>
<td>0.504444</td>
<td>0.375556</td>
<td>0.552778</td>
<td>0.558056</td>
<td>0.551806</td>
<td>0.67383</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.365968</td>
<td>0.325311</td>
<td>0.502351</td>
<td>0.550484</td>
<td>0.55574</td>
<td>0.549516</td>
<td>0.549217</td>
<td>0.680217</td>
</tr>
<tr>
<td>MCC</td>
<td>0.324728</td>
<td>0.455355</td>
<td>0.332324</td>
<td>0.455909</td>
<td>0.439796</td>
<td>0.41512</td>
<td>0.428936</td>
<td>0.52882</td>
</tr>
<tr>
<td>NPV</td>
<td>0.680775</td>
<td>0.660581</td>
<td>0.749101</td>
<td>0.684924</td>
<td>0.773167</td>
<td>0.775795</td>
<td>0.772683</td>
<td>0.849793</td>
</tr>
<tr>
<td>FPR</td>
<td>0.316389</td>
<td>0.336667</td>
<td>0.247778</td>
<td>0.312222</td>
<td>0.223611</td>
<td>0.220972</td>
<td>0.224097</td>
<td>0.153137</td>
</tr>
<tr>
<td>FNR</td>
<td>0.638513</td>
<td>0.676033</td>
<td>0.499725</td>
<td>0.627548</td>
<td>0.451791</td>
<td>0.446556</td>
<td>0.452755</td>
<td>0.312667</td>
</tr>
</tbody>
</table>

Table 12: Evaluation of the proposed work trained with existing Feature set (Statistical Features, EMA, DEMA, Existing weighted Holoentropy, Percentile, and Moment) for dataset 2 corresponding TO $K=5$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.381818</td>
<td>0.472176</td>
<td>0.32562</td>
<td>0.332231</td>
<td>0.551515</td>
<td>0.551653</td>
<td>0.541185</td>
<td>0.681996</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.6925</td>
<td>0.730856</td>
<td>0.664167</td>
<td>0.6675</td>
<td>0.770856</td>
<td>0.778125</td>
<td>0.772847</td>
<td>0.844607</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.588366</td>
<td>0.648938</td>
<td>0.550693</td>
<td>0.55125</td>
<td>0.702124</td>
<td>0.702216</td>
<td>0.695199</td>
<td>0.79235</td>
</tr>
<tr>
<td>Precision</td>
<td>0.385</td>
<td>0.476111</td>
<td>0.328333</td>
<td>0.335</td>
<td>0.556111</td>
<td>0.55625</td>
<td>0.545694</td>
<td>0.66712</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.383422</td>
<td>0.474136</td>
<td>0.326971</td>
<td>0.33361</td>
<td>0.553804</td>
<td>0.553942</td>
<td>0.54343</td>
<td>0.674555</td>
</tr>
<tr>
<td>MCC</td>
<td>0.365465</td>
<td>0.440161</td>
<td>0.482189</td>
<td>0.312658</td>
<td>0.459247</td>
<td>0.448974</td>
<td>0.459967</td>
<td>0.552874</td>
</tr>
<tr>
<td>NPV</td>
<td>0.689627</td>
<td>0.734993</td>
<td>0.661141</td>
<td>0.66473</td>
<td>0.774827</td>
<td>0.774896</td>
<td>0.76964</td>
<td>0.847696</td>
</tr>
<tr>
<td>FPR</td>
<td>0.3075</td>
<td>0.261944</td>
<td>0.335833</td>
<td>0.3325</td>
<td>0.221944</td>
<td>0.221875</td>
<td>0.227153</td>
<td>0.155393</td>
</tr>
<tr>
<td>FNR</td>
<td>0.618182</td>
<td>0.527824</td>
<td>0.67438</td>
<td>0.667769</td>
<td>0.448485</td>
<td>0.448347</td>
<td>0.458815</td>
<td>0.318004</td>
</tr>
</tbody>
</table>

9.5 Performance Analysis of Proposed Work: Proposed Feature Set

The performance of the proposed work trained with the proposed feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) is compared over the existing works like SGM-CNN[1], NN, CNN, RNN, MULTI-CLASSIFIER+WOA, MULTI-CLASSIFIER+MFO, MULTI-CLASSIFIER+SOA trained with the same proposed feature set (Statistical Features, EMA, DEMA, IWH, Percentile and Moment). This evaluation is carried out in terms of “Accuracy, Sensitivity, Specificity, Precision, F-measure, FDR, FNR, FPR, NPV, and MCC” respectively for varying k-values. This assessment is undergone for dataset1 (CICIDS2018) and dataset2 (UNSWNB15), respectively. The results acquired for dataset-1 are shown in Table XIII to Table XVIII respectively. In addition, the results acquired for Dataset 2 are shown in Tables XIX to XXIII. On observing the outcomes, the proposed work trained with the proposed feature set has attained the highest performance in the case of both datasets. This is owing to the introduction of a new IWH-based feature extraction mechanism, wherein the weight functions are computed for the input, rather than the standard value used in the standard holoentropy model. Moreover, the accuracy of the proposed work trained with the proposed feature set for dataset 1 corresponding to $K=1$ is 0.932433, which is the best value compared to the accuracy value of the proposed work trained with the existing feature set (0.896598). For dataset 1 corresponding to $K=2$, the proposed work trained with the proposed feature set had attained the accuracy value of 0.934932, which accuracy of the proposed work trained with the existing feature set 0.793213. Thus, from the evaluation, it is clear that the proposed work trained with the proposed feature set is much more efficient in detecting network intrusions.
A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

Table 14: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset I corresponding to $K = 1$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.710744</td>
<td>0.328926</td>
<td>0.827548</td>
<td>0.32562</td>
<td>0.773967</td>
<td>0.779367</td>
<td>0.769972</td>
<td>0.871877</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.858333</td>
<td>0.656833</td>
<td>0.917222</td>
<td>0.664167</td>
<td>0.890208</td>
<td>0.890208</td>
<td>0.888194</td>
<td>0.955761</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.808864</td>
<td>0.552909</td>
<td>0.887165</td>
<td>0.550693</td>
<td>0.851247</td>
<td>0.851247</td>
<td>0.848569</td>
<td>0.932433</td>
</tr>
<tr>
<td>Precision F-Measure</td>
<td>0.716667</td>
<td>0.331667</td>
<td>0.834444</td>
<td>0.328333</td>
<td>0.780417</td>
<td>0.780417</td>
<td>0.776389</td>
<td>0.88815</td>
</tr>
<tr>
<td>MCC</td>
<td>0.570257</td>
<td>0.305857</td>
<td>0.746314</td>
<td>0.365693</td>
<td>0.610707</td>
<td>0.617735</td>
<td>0.658654</td>
<td>0.861967</td>
</tr>
<tr>
<td>NPV</td>
<td>0.854772</td>
<td>0.663071</td>
<td>0.913416</td>
<td>0.661411</td>
<td>0.886515</td>
<td>0.885615</td>
<td>0.884509</td>
<td>0.948727</td>
</tr>
<tr>
<td>FPR</td>
<td>0.141667</td>
<td>0.334167</td>
<td>0.08278</td>
<td>0.335833</td>
<td>0.107083</td>
<td>0.107153</td>
<td>0.107153</td>
<td>0.044239</td>
</tr>
<tr>
<td>FNR</td>
<td>0.289256</td>
<td>0.671074</td>
<td>0.172452</td>
<td>0.67438</td>
<td>0.226033</td>
<td>0.226033</td>
<td>0.230028</td>
<td>0.128123</td>
</tr>
</tbody>
</table>

Table 15: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset I corresponding to $K = 2$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.682645</td>
<td>0.801102</td>
<td>0.812672</td>
<td>0.844628</td>
<td>0.779339</td>
<td>0.779201</td>
<td>0.779201</td>
<td>0.876952</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.844167</td>
<td>0.903889</td>
<td>0.909722</td>
<td>0.925833</td>
<td>0.892917</td>
<td>0.892847</td>
<td>0.892847</td>
<td>0.957446</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.790028</td>
<td>0.869437</td>
<td>0.877193</td>
<td>0.898615</td>
<td>0.854848</td>
<td>0.854755</td>
<td>0.854755</td>
<td>0.934932</td>
</tr>
<tr>
<td>Precision F-Measure</td>
<td>0.688333</td>
<td>0.807778</td>
<td>0.819444</td>
<td>0.851667</td>
<td>0.785833</td>
<td>0.785894</td>
<td>0.785894</td>
<td>0.892979</td>
</tr>
<tr>
<td>MCC</td>
<td>0.685477</td>
<td>0.804426</td>
<td>0.816044</td>
<td>0.848133</td>
<td>0.782573</td>
<td>0.782434</td>
<td>0.782434</td>
<td>0.884814</td>
</tr>
<tr>
<td>NPV</td>
<td>0.840664</td>
<td>0.906138</td>
<td>0.905947</td>
<td>0.921992</td>
<td>0.889212</td>
<td>0.889142</td>
<td>0.889142</td>
<td>0.950495</td>
</tr>
<tr>
<td>FPR</td>
<td>0.155833</td>
<td>0.096111</td>
<td>0.090278</td>
<td>0.074167</td>
<td>0.107083</td>
<td>0.107153</td>
<td>0.107153</td>
<td>0.042554</td>
</tr>
<tr>
<td>FNR</td>
<td>0.317355</td>
<td>0.198898</td>
<td>0.187328</td>
<td>0.155372</td>
<td>0.220661</td>
<td>0.220799</td>
<td>0.220799</td>
<td>0.123048</td>
</tr>
</tbody>
</table>

Table 15: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset I corresponding to $K = 3$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.788981</td>
<td>0.627548</td>
<td>0.816529</td>
<td>0.743802</td>
<td>0.774793</td>
<td>0.774793</td>
<td>0.767631</td>
<td>0.872373</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.897778</td>
<td>0.816389</td>
<td>0.911667</td>
<td>0.875</td>
<td>0.890625</td>
<td>0.890625</td>
<td>0.887014</td>
<td>0.955056</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.861311</td>
<td>0.750393</td>
<td>0.879778</td>
<td>0.831025</td>
<td>0.851801</td>
<td>0.851801</td>
<td>0.846999</td>
<td>0.931779</td>
</tr>
<tr>
<td>Precision F-Measure</td>
<td>0.795556</td>
<td>0.632778</td>
<td>0.823333</td>
<td>0.75</td>
<td>0.78125</td>
<td>0.78125</td>
<td>0.774028</td>
<td>0.883335</td>
</tr>
<tr>
<td>MCC</td>
<td>0.688182</td>
<td>0.444857</td>
<td>0.729705</td>
<td>0.620084</td>
<td>0.600587</td>
<td>0.589589</td>
<td>0.646672</td>
<td>0.807347</td>
</tr>
<tr>
<td>NPV</td>
<td>0.894053</td>
<td>0.813001</td>
<td>0.907884</td>
<td>0.871369</td>
<td>0.886929</td>
<td>0.886929</td>
<td>0.883333</td>
<td>0.948124</td>
</tr>
<tr>
<td>FPR</td>
<td>0.102222</td>
<td>0.183611</td>
<td>0.088333</td>
<td>0.125</td>
<td>0.109375</td>
<td>0.109375</td>
<td>0.112986</td>
<td>0.044944</td>
</tr>
<tr>
<td>FNR</td>
<td>0.211019</td>
<td>0.372452</td>
<td>0.183471</td>
<td>0.256198</td>
<td>0.225207</td>
<td>0.225207</td>
<td>0.232369</td>
<td>0.127627</td>
</tr>
</tbody>
</table>
### Table 16: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset 1 corresponding to $K=4$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.75978</td>
<td>0.808815</td>
<td>0.82259</td>
<td>0.472727</td>
<td>0.778375</td>
<td>0.74105</td>
<td>0.774242</td>
<td>0.876174</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.883056</td>
<td>0.907778</td>
<td>0.914722</td>
<td>0.738333</td>
<td>0.892431</td>
<td>0.890278</td>
<td>0.890347</td>
<td>0.951776</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.841736</td>
<td>0.874608</td>
<td>0.883841</td>
<td>0.649307</td>
<td>0.854201</td>
<td>0.851339</td>
<td>0.851431</td>
<td>0.934536</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.766111</td>
<td>0.815556</td>
<td>0.829444</td>
<td>0.476667</td>
<td>0.784861</td>
<td>0.780556</td>
<td>0.780694</td>
<td>0.892236</td>
</tr>
<tr>
<td>MCC</td>
<td>0.644167</td>
<td>0.718078</td>
<td>0.73884</td>
<td>0.460451</td>
<td>0.77317</td>
<td>0.777455</td>
<td>0.777455</td>
<td>0.884052</td>
</tr>
<tr>
<td>NPV</td>
<td>0.879391</td>
<td>0.804011</td>
<td>0.910927</td>
<td>0.73527</td>
<td>0.888728</td>
<td>0.886584</td>
<td>0.886653</td>
<td>0.950213</td>
</tr>
<tr>
<td>FPR</td>
<td>0.24022</td>
<td>0.191185</td>
<td>0.17741</td>
<td>0.527273</td>
<td>0.221625</td>
<td>0.225895</td>
<td>0.225758</td>
<td>0.123826</td>
</tr>
</tbody>
</table>

### Table 17: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset 1 corresponding to $K=5$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.699725</td>
<td>0.80832</td>
<td>0.81157</td>
<td>0.808264</td>
<td>0.782369</td>
<td>0.782369</td>
<td>0.77741</td>
<td>0.879728</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.852778</td>
<td>0.845278</td>
<td>0.909167</td>
<td>0.9075</td>
<td>0.894444</td>
<td>0.894444</td>
<td>0.891944</td>
<td>0.958288</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.801477</td>
<td>0.892447</td>
<td>0.876454</td>
<td>0.87438</td>
<td>0.856879</td>
<td>0.856879</td>
<td>0.853555</td>
<td>0.936219</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.706556</td>
<td>0.855556</td>
<td>0.818333</td>
<td>0.815</td>
<td>0.788889</td>
<td>0.788889</td>
<td>0.783889</td>
<td>0.895587</td>
</tr>
<tr>
<td>MCC</td>
<td>0.553647</td>
<td>0.83019</td>
<td>0.722231</td>
<td>0.717248</td>
<td>0.606465</td>
<td>0.651928</td>
<td>0.639113</td>
<td>0.894271</td>
</tr>
<tr>
<td>NPV</td>
<td>0.849239</td>
<td>0.894136</td>
<td>0.905394</td>
<td>0.903734</td>
<td>0.890733</td>
<td>0.888243</td>
<td>0.888243</td>
<td>0.951393</td>
</tr>
<tr>
<td>FPR</td>
<td>0.147222</td>
<td>0.054722</td>
<td>0.090833</td>
<td>0.0925</td>
<td>0.105556</td>
<td>0.105556</td>
<td>0.108056</td>
<td>0.041712</td>
</tr>
<tr>
<td>FNR</td>
<td>0.24022</td>
<td>0.191185</td>
<td>0.17741</td>
<td>0.527273</td>
<td>0.221625</td>
<td>0.225895</td>
<td>0.225758</td>
<td>0.123826</td>
</tr>
</tbody>
</table>

### Table 18: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset 2 corresponding to $K=1$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.530028</td>
<td>0.373003</td>
<td>0.511846</td>
<td>0.310744</td>
<td>0.595317</td>
<td>0.595041</td>
<td>0.591598</td>
<td>0.753959</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.767222</td>
<td>0.688056</td>
<td>0.75056</td>
<td>0.656667</td>
<td>0.800139</td>
<td>0.798264</td>
<td>0.798264</td>
<td>0.88132</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.687719</td>
<td>0.582456</td>
<td>0.675531</td>
<td>0.54072</td>
<td>0.731487</td>
<td>0.731302</td>
<td>0.728994</td>
<td>0.839238</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.532227</td>
<td>0.37455</td>
<td>0.51397</td>
<td>0.312033</td>
<td>0.597877</td>
<td>0.59751</td>
<td>0.594053</td>
<td>0.757849</td>
</tr>
<tr>
<td>MCC</td>
<td>0.368424</td>
<td>0.485787</td>
<td>0.376506</td>
<td>0.359878</td>
<td>0.523316</td>
<td>0.520986</td>
<td>0.543582</td>
<td>0.68397</td>
</tr>
<tr>
<td>NPV</td>
<td>0.764039</td>
<td>0.685201</td>
<td>0.75491</td>
<td>0.653942</td>
<td>0.796819</td>
<td>0.79668</td>
<td>0.794982</td>
<td>0.87734</td>
</tr>
<tr>
<td>FPR</td>
<td>0.232778</td>
<td>0.311944</td>
<td>0.241944</td>
<td>0.343333</td>
<td>0.199861</td>
<td>0.2</td>
<td>0.201736</td>
<td>0.11868</td>
</tr>
<tr>
<td>FNR</td>
<td>0.469972</td>
<td>0.626997</td>
<td>0.488154</td>
<td>0.689256</td>
<td>0.404683</td>
<td>0.404959</td>
<td>0.408402</td>
<td>0.246041</td>
</tr>
</tbody>
</table>
### Table 19: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset 2 corresponding to $K = 2$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.489256</td>
<td>0.332782</td>
<td>0.497521</td>
<td>0.327824</td>
<td>0.586777</td>
<td>0.587603</td>
<td>0.586777</td>
<td>0.747891</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.746667</td>
<td>0.667778</td>
<td>0.750833</td>
<td>0.665278</td>
<td>0.795833</td>
<td>0.79625</td>
<td>0.795833</td>
<td>0.875717</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.660388</td>
<td>0.555494</td>
<td>0.665928</td>
<td>0.55217</td>
<td>0.725762</td>
<td>0.726316</td>
<td>0.725762</td>
<td>0.835447</td>
</tr>
<tr>
<td>Precision</td>
<td>0.493333</td>
<td>0.335556</td>
<td>0.505167</td>
<td>0.330556</td>
<td>0.591667</td>
<td>0.5925</td>
<td>0.591667</td>
<td>0.756035</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.491286</td>
<td>0.334163</td>
<td>0.495555</td>
<td>0.329184</td>
<td>0.589212</td>
<td>0.590041</td>
<td>0.589212</td>
<td>0.751938</td>
</tr>
<tr>
<td>MCC</td>
<td>0.305657</td>
<td>0.396763</td>
<td>0.315754</td>
<td>0.321822</td>
<td>0.517907</td>
<td>0.506909</td>
<td>0.527958</td>
<td>0.671151</td>
</tr>
<tr>
<td>NPV</td>
<td>0.743568</td>
<td>0.665007</td>
<td>0.747718</td>
<td>0.662517</td>
<td>0.792531</td>
<td>0.792946</td>
<td>0.792531</td>
<td>0.874892</td>
</tr>
<tr>
<td>FPR</td>
<td>0.253333</td>
<td>0.332222</td>
<td>0.249167</td>
<td>0.334722</td>
<td>0.204167</td>
<td>0.20375</td>
<td>0.204167</td>
<td>0.121483</td>
</tr>
<tr>
<td>FNR</td>
<td>0.510744</td>
<td>0.667218</td>
<td>0.502479</td>
<td>0.672176</td>
<td>0.413223</td>
<td>0.412397</td>
<td>0.413223</td>
<td>0.252109</td>
</tr>
</tbody>
</table>

### Table 20: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset 2 corresponding to $K = 3$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.505234</td>
<td>0.338843</td>
<td>0.513499</td>
<td>0.326722</td>
<td>0.598072</td>
<td>0.597796</td>
<td>0.597394</td>
<td>0.756804</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.754722</td>
<td>0.670833</td>
<td>0.758889</td>
<td>0.664722</td>
<td>0.801528</td>
<td>0.801389</td>
<td>0.801528</td>
<td>0.883093</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.671099</td>
<td>0.559557</td>
<td>0.676639</td>
<td>0.551431</td>
<td>0.733333</td>
<td>0.733149</td>
<td>0.733241</td>
<td>0.841512</td>
</tr>
<tr>
<td>Precision</td>
<td>0.509444</td>
<td>0.341667</td>
<td>0.517778</td>
<td>0.329444</td>
<td>0.603056</td>
<td>0.602778</td>
<td>0.602917</td>
<td>0.76514</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.507331</td>
<td>0.340249</td>
<td>0.515629</td>
<td>0.328077</td>
<td>0.605053</td>
<td>0.600277</td>
<td>0.600415</td>
<td>0.769947</td>
</tr>
<tr>
<td>MCC</td>
<td>0.464472</td>
<td>0.417056</td>
<td>0.375705</td>
<td>0.407199</td>
<td>0.498984</td>
<td>0.511917</td>
<td>0.478388</td>
<td>0.64223</td>
</tr>
<tr>
<td>NPV</td>
<td>0.751591</td>
<td>0.66805</td>
<td>0.75574</td>
<td>0.661964</td>
<td>0.798202</td>
<td>0.798064</td>
<td>0.798133</td>
<td>0.879433</td>
</tr>
<tr>
<td>FPR</td>
<td>0.245278</td>
<td>0.329167</td>
<td>0.241111</td>
<td>0.335278</td>
<td>0.198472</td>
<td>0.198611</td>
<td>0.198542</td>
<td>0.116907</td>
</tr>
<tr>
<td>FNR</td>
<td>0.494766</td>
<td>0.661157</td>
<td>0.486501</td>
<td>0.673278</td>
<td>0.401928</td>
<td>0.402204</td>
<td>0.402066</td>
<td>0.243196</td>
</tr>
</tbody>
</table>

### Table 21: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset 2 corresponding to $K = 4$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.530028</td>
<td>0.324518</td>
<td>0.507989</td>
<td>0.332782</td>
<td>0.599174</td>
<td>0.596419</td>
<td>0.598623</td>
<td>0.759461</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.767222</td>
<td>0.66311</td>
<td>0.756111</td>
<td>0.667778</td>
<td>0.802083</td>
<td>0.800694</td>
<td>0.801806</td>
<td>0.88435</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.687719</td>
<td>0.549954</td>
<td>0.672946</td>
<td>0.555494</td>
<td>0.734072</td>
<td>0.732225</td>
<td>0.733703</td>
<td>0.843204</td>
</tr>
<tr>
<td>Precision</td>
<td>0.534444</td>
<td>0.327222</td>
<td>0.512222</td>
<td>0.335556</td>
<td>0.604167</td>
<td>0.601389</td>
<td>0.603611</td>
<td>0.767702</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.532227</td>
<td>0.325864</td>
<td>0.510097</td>
<td>0.334163</td>
<td>0.601666</td>
<td>0.598893</td>
<td>0.601107</td>
<td>0.763557</td>
</tr>
<tr>
<td>MCC</td>
<td>0.385805</td>
<td>0.454716</td>
<td>0.304809</td>
<td>0.315268</td>
<td>0.503134</td>
<td>0.541694</td>
<td>0.518478</td>
<td>0.694757</td>
</tr>
<tr>
<td>NPV</td>
<td>0.764039</td>
<td>0.668588</td>
<td>0.752974</td>
<td>0.665007</td>
<td>0.789755</td>
<td>0.797372</td>
<td>0.798479</td>
<td>0.880701</td>
</tr>
<tr>
<td>FPR</td>
<td>0.232778</td>
<td>0.336389</td>
<td>0.243889</td>
<td>0.332222</td>
<td>0.197917</td>
<td>0.199306</td>
<td>0.198194</td>
<td>0.11565</td>
</tr>
<tr>
<td>FNR</td>
<td>0.469972</td>
<td>0.675482</td>
<td>0.492011</td>
<td>0.667218</td>
<td>0.400826</td>
<td>0.403581</td>
<td>0.401377</td>
<td>0.240539</td>
</tr>
</tbody>
</table>
Table 22: Evaluation of the proposed work trained with the proposed Feature set (Statistical Features, EMA, DEMA, IWH, Percentile, and Moment) for dataset 2 corresponding to \( K = 5 \)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.52011</td>
<td>0.363085</td>
<td>0.512948</td>
<td>0.32011</td>
<td>0.588567</td>
<td>0.585262</td>
<td>0.59022</td>
<td>0.751886</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.762222</td>
<td>0.683056</td>
<td>0.758611</td>
<td>0.661389</td>
<td>0.796736</td>
<td>0.795069</td>
<td>0.797569</td>
<td>0.879418</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.681071</td>
<td>0.575808</td>
<td>0.67627</td>
<td>0.546999</td>
<td>0.726962</td>
<td>0.724746</td>
<td>0.72807</td>
<td>0.836957</td>
</tr>
<tr>
<td>Precision</td>
<td>0.524444</td>
<td>0.366111</td>
<td>0.517222</td>
<td>0.322778</td>
<td>0.593472</td>
<td>0.590139</td>
<td>0.595139</td>
<td>0.75841</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.522268</td>
<td>0.364592</td>
<td>0.515076</td>
<td>0.321438</td>
<td>0.59101</td>
<td>0.58769</td>
<td>0.592669</td>
<td>0.755134</td>
</tr>
<tr>
<td>MCC</td>
<td>0.385089</td>
<td>0.422662</td>
<td>0.496873</td>
<td>0.409297</td>
<td>0.509142</td>
<td>0.522074</td>
<td>0.501478</td>
<td>0.658644</td>
</tr>
<tr>
<td>NPV</td>
<td>0.759059</td>
<td>0.680221</td>
<td>0.755463</td>
<td>0.658645</td>
<td>0.79343</td>
<td>0.79177</td>
<td>0.79426</td>
<td>0.87606</td>
</tr>
<tr>
<td>FPR</td>
<td>0.237778</td>
<td>0.316944</td>
<td>0.241389</td>
<td>0.38611</td>
<td>0.203264</td>
<td>0.204931</td>
<td>0.202431</td>
<td>0.120582</td>
</tr>
<tr>
<td>FNR</td>
<td>0.47989</td>
<td>0.636915</td>
<td>0.487052</td>
<td>0.67989</td>
<td>0.411433</td>
<td>0.414738</td>
<td>0.40978</td>
<td>0.248114</td>
</tr>
</tbody>
</table>

10. Conclusion

This paper proposed an intrusion detection model for class imbalance data by following four major phases: (a) Pre-processing, (b) Imbalance processing, (c) Feature extraction, and (d) Intrusion detection phase. Initially, the collected class imbalance data was pre-processed via data cleaning and data standardization approach. Then, the pre-processed class imbalance data was more balanced in the imbalanced processing phase by a new improved over-sampling technique using SMOTE and the Multi-kernel FCM clustering model. Subsequently, multi-features like EMA, DEMA, IWH, and statistical features (Mean, Median, Standard deviation, percentile, and moment) were extracted from these balanced data. The extracted overall features are given for K-fold validation. Then, the intrusion detection phase was modeled with a combination of four individual ML models such as SVM, optimized DBN with incremental learning, NB, and RF to achieve the utmost detection performance. Each of the classifiers is trained with the acquired K-fold data features. Moreover, to achieve the utmost detection accuracy as well as better tradeoff performance, the weight of DBN was fine-tuned via a new SI-SOA, which was an improved version of standard SOA. Then, the logoff performance was computed from the outcome acquired from each of the individual classifiers (SVM, DBN, NB, and RF). Finally, the log computed result will portray the detected attacks in the network. The performance of the proposed work (MULTI-CASSIFIER+SI-SOA) is compared over the existing models like SGM-CNN[1], NN, CNN, RNN, MULTI-CASSIFIER+WOA, MULTI-CASSIFIER+MFO, MULTI-CASSIFIER+SOA respectively. In this research work, we have fixed the K-value of K-fold evaluation as 5 (K = 5), and we've varied this K value from 1, 2, 3, 4, and 5, to evaluate the performance of the projected model. Furthermore, the performance was based on the different performance measures including "accuracy, sensitivity, specificity, precision, F-measure, FDR, FNR, FPR, NPV, and MCC". The accuracy of the projected model is higher for every variation in the K-values. At K = 5, the proposed work attained the highest accuracy value as 92%, which is 13%, 9.7%, 11.4%, 11.5%, 11.6%, 11.6% and 11.8% better than the accuracy value recorded by the existing models like SGM-CNN, NN, CNN, RNN, MULTI-CASSIFIER+WOA, MULTI-CASSIFIER+MFO, MULTI-CASSIFIER+SOA respectively.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach


A Combined Machine Learning Model for Intrusion Detection in Imbalanced Dataset: A Hybrid Optimization-Incremental Learning Approach

(IAEAC), Chengdu, China, 2019, pp. 982-986.
doi: 10.1109/IAEAC47372.2019.8997825


[31] Dataset2, from: "https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html", access date: 2021-08-10


[38] EMA from: "https://en.wikipedia.org/wiki/Moving_average"


