Article

A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU

Sweta Bhattacharya [1], Siva Rama Krishnan S [1], Praveen Kumar Reddy Maddikunta [1], Rajesh Kaluri [1], Saurabh Singh [2], Thippa Reddy Gadekallu [1,*], Mamoun Alazab [3,*] and Usman Tariq [4,*]

1   School of Information Technology and Engineering, VIT - Vellore, Tamil Nadu 632014, India;
    sweta.b@vit.ac.in (S.B.); siva.s@vit.ac.in (S.R.K.S.); praveenkumarreddy@vit.ac.in (P.K.R.M.);
    rajesh.kaluri@vit.ac.in (R.K.)
2   School of Computer, Information and Communication Engineering, Kunsan National University,
    Gunsan 54150, Korea; singh1989@jbnu.ac.kr
3   Senior IEEE Member, IT and Environment, Charles Darwin University, 0815 Darwin, Australia
4   College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University,
    Al-Khraj 11942, Saudi Arabia
*   Correspondence: thippareddy.g@vit.ac.in (T.R.G.); mamoun.alazab@cdu.edu.au (M.A.);
    u.tariq@psau.edu.sa (U.T.)

check for updates

Abstract: The enormous popularity of the internet across all spheres of human life has introduced various risks of malicious attacks in the network. The activities performed over the network could be effortlessly proliferated, which has led to the emergence of intrusion detection systems. The patterns of the attacks are also dynamic, which necessitates efficient classification and prediction of cyber attacks. In this paper we propose a hybrid principal component analysis (PCA)-firefly based machine learning model to classify intrusion detection system (IDS) datasets. The dataset used in the study is collected from Kaggle. The model first performs One-Hot encoding for the transformation of the IDS datasets. The hybrid PCA-firefly algorithm is then used for dimensionality reduction. The XGBoost algorithm is implemented on the reduced dataset for classification. A comprehensive evaluation of the model is conducted with the state of the art machine learning approaches to justify the superiority of our proposed approach. The experimental results confirm the fact that the proposed model performs better than the existing machine learning models.

Keywords: intrusion detection system; principal component analysis (PCA); firefly; XGBoost; One-Hot encoding; machine learning; Google Colab GPU

1. Introduction

Life without internet has almost become impossible in the present day and age. The potential of internet is enormous and its growth has been reflected in the functioning of business models pertaining to education, entertainment, healthcare and all the various types of organizations around the world. However, use of internet in every sphere of human life has its own challenges. The most significant challenge deals with data security. Network intrusion is a situation of security breach due to unauthorized access in a computational network. The process of identifying the different types of intrusions in a network is performed by an intrusion detection system (IDS) [1]. The attacks in the IDS can be categorized as Probe attack, DoS attack, R2L attack, U2R attack. In probe attack, the unauthorized personnel 'sniffs' the network and identifies the vulnerabilities of a particular target

resource. As an example, the attacker can use an uncommon port number as the identification with other IP addresses to attack on different ports. The DoS attacks are targeted to make the network resources unavailable by flooding unnecessary data packets by the attacker. In this case, an internal attack can be generated where the user targets a particular network resource and floods it with internet control message protocol (ICMP) by using a simple command "ping -t". A U2R attack is an attack where the attacker gains access to the root privileges of the network. This leads to several disasters such as gaining unauthorized access to the a control list where the users can modify permissions as per their convenience. In R2L attack the attacker gets unauthorized access into the victim's network thereby sniffing of the data gets easier, which can be prevented by having a virtual private network (VPN) framework [2]. In order to combat these network attacks, there is a dire need for deploying efficient IDS, which acts as a surveillance system for detecting anomalies in the incoming network traffic. Any malicious activity in the network is immediately reported to the network operation center (NOC). The IDS can be deployed either in the network or in a host. Based on the deployment type the IDS can be categorised as network IDS or host IDS. The network IDS is implemented in the border router as well as in various subnets to detect any abnormal behaviour in the network traffic. Such abnormalities are recorded as logs in the IDS servers to prevent similar attacks on the network perimeter. The host IDS, on the contrary, is deployed on each individual host and the incoming and outgoing network traffic is monitored for abnormalities [3–5]. The existing data is compared with the previous data log and any discrepancies detected are reported to the administrator.

IDS systems can be hardware as well as software oriented wherein the software based IDS is far more convenient and configurable than the hardware based IDS, which faces issues in handling data traffic requiring rigorous maintenance tasks. In spite of such issues research on hardware based IDS has gained immense momentum for the detection of attacks. The emphasis hence has been more on the performance of advanced graphics processors, which provide higher performance in the detection of attacks using hardware based IDS.

Since any network is extremely vulnerable to such attacks, organizations need to have efficient defensive mechanisms installed and deployed to identify the maximum possible threats. NIDS and HIDS sensors are resource expensive and impractical to be deployed covering the entire network. Inability to strategically optimize the usage of IDS could lead to important resources being exposed to adversarial attacks. Any IDS framework comes bundled with pre-defined signatures, which alert to any anomalies in the network traffic. However this framework needs to be customized to adhere to all possible organizational needs. Moreover, investigation of IDS alerts tend to be extremely resource intensive, and might require additional information from other network tools to confirm and decide on the seriousness of the alerts generated. These decisions require security expert personnel capable of the interpretation of system outputs and computing of crucial functions. Hence experienced security experts are required for the remediation, detection and management of such threats [3–5].

As the dataset used in this work has huge dimension, an effective dimensionality reduction mechanism had to be developed to reduce the burden of the classifier. Moreover, a dimensionality reduction mechanism would facilitate the classifier to choose the most important attributes and eliminate the attributes that have negative impact on the performance of the classifiers. This motivated us to design a model making use of a PCA based firefly algorithm to effectively choose the most relevant attributes from the IDS dataset. The features selected from this hybrid PCA-firefly algorithm are therefore trained using the XGBoost classifier. The performance of the proposed model is then evaluated with existing algorithms.

The rest of the paper is organized as follows: Section 2 presents related work, Section 3 discusses briefly the background algorithms used in this work, Section 4 discribes the proposed methodology, Section 5 highlights the results of exprimentation and Section 6 incorporates the conclusion and points out the direction of future works.

2. Related Work

Intrusion detection is a major task of any network security tool. The various intrusion detection and prevention systems deployed in networks have associated performance and efficiency issues. The performance of the IDS depends on its accuracy in detecting network anomalies with decreased number of false positive alarms being generated. Various researchers have worked to resolve such performance issues by implementing various machine learning approaches on IDS datasets [6]. Support vector machines (SVM), multi-layer perceptron network and various other ML techniques have been used, each with limitations pertinent to the handling of large network datasets. Researchers have proposed the use of classification techniques to eradicate such accuracy and performance issues in the prediction of malicious network activities. Apart from the popular machine learning techniques extreme learning machine (ELM) and NSL knowledge discovery data mining have been identified as a standard for the evaluation of intrusion detection mechanisms in the network [7]. Researchers have also implemented random forest classifier on the IDS dataset sample. Such approaches have helped in the experimentation of datasets and analysing effects of malicious attacks considering various perspectives and dimensions [8].

It is well known that applications of machine learning has a successful track record of automatically detecting and classifying intrusions both at network and host level in the quickest time frame. However, it is also essential to consider the volatility and ever changing characteristics of the malicious attacks involving larger volumes of stake holders. This issue needs to be resolved providing scalability. Moreover, the available datasets require continuous updates based on the dynamic characteristics of malware attacks. Several researchers have suggested the implementation of deep learning models and deep neural networks (DNN) for the development of flexible and dynamic IDS that would be capable of efficiently detecting and classifying capricious network attacks [9–11]. The results of the DNN model reveal potential of high dimensional feature representation of the IDS data when fed into hidden layers in comparison to the other machine learning approaches. Based on the performance of the DNN model, the authors have proposed a scalable hybrid IDS-AlertNet system that would efficiently monitor network traffic and proactively send alert on cyber attacks. In coordination with the use of DNN, convolutional neural network algorithms have also been identified as an advanced and superior technology for extraction of features in an intrusion dataset for the classification purposes [12].

There exists almost eleven datasets namely DARPA, KDD99, ISC2012 and ADFA13, which have been used for classification and analysis of intrusion datasets. However, most of these datasets fail to incorporate network traffic diversity, volume and versatility related information on malicious attacks. Moreover, there are cases of anonymous packet information, insufficient payload information, lack of features and metadata [13]. The classical machine learning algorithms when applied to the biased public and biased datasets fail to yield accurate results making them impractical to be used in real-time situations. The authors hence have split the datasets in a disjoint fashion across multiple time scales for the purpose of training and testing of the ML model. Image processing techniques in combination with optimal parameterization and deep learning models have helped to enhance robustness in the network and eradicate malwares completely. Since the approach provides visual detection of network intrusion it is justified as a real time solution for the deployment of intrusion detection system [14].

Newer attacks pop up every day and identifying these attacks is a major challenge. Once the attacks have been identified, the IDS have to be fed in with appropriate responses, which constitute data gathering, feature selection and decision system. Deep learning is said to be one of the best approaches, which can be used in IDS due to the fact that its training duration is less and accuracy is better. The authors in [15] survey different deep learning approaches incorporated in IDS and present a comparative analysis of the same.

Burstiness of data has been given a huge importance to information security so that the value of data can be used for business intelligence. The speed at, which the data is generated every second has made detection of attacks a challenging task. The authors in [16] have used Spark-Chi-SVM technique for intrusion detection. The authors also use ChiSqSelector for feature selection and a built-in intrusion

detection technique is used with the help of support vector machine (SVM) classifier on Apache Spark Big Data platform. The authors conclude that Spark-Chi-SVM technique offers higher performance by reducing the training duration and is also effective for Big data scenarios.

Machine and deep learning techniques have been significantly implemented for IDS in wireless sensor network (WSN). For observing the major network resources in WSN, the researchers implement Boltzmann machine-based clustered IDS (RBC-IDS), which is a DL-based IDS technique. In the study [17], researchers analyse the performance of RBC-IDS, compare it to the existing adaptive machine learning-based IDS and conclude that the detection duration of RBC-IDS is roughly twice than the ASCH-IDS.

Several researchers have used nature-inspired optimization algorithms along with machine learning algorithms for classifying the datasets. Advantages of using nature inspired algorithms is that, they can help the classifiers in overcoming the problem of getting struck at local minima. The authors in [18–21] propose several hybrid classification models using nature inspired algorithms like cuckoo search, BAT, firefly, genetic algorithms etc, to classify diabetes and heart disease datasets. The authors [22] compared RIFCM algorithm with existing algorithms for the suitability in analyzing satellite images. In [23] the authors proved a novel algorithm to cluster categorical data with rough set theory. A new algorithm based on rough sets on fuzzy approximation and intuitionistic fuzzy approximation is proposed by the authors in [24]. The authors in [25] proposed a hybrid intuitionistic fuzzy and rough set-based approach for the classification of breast cancer dataset.

Most of the studies conducted in this area have primarily focused on application of optimized machine learning and neural network approaches for intrusion detection. However, the optimum level of accuracy has not been achieved due to their narrow focus and emphasis only on the application techniques and failure to consider feature engineering. In order to fulfill this objective, in our present study we have used PCA and firefly to choose the most significant features eliminating irrelevant ones, which have negative effect on the accuracy of the prediction.

3. Background

XGBoost: XGBoost is an optimized gradient tree boosting system that creates decision trees in a sequential form [26]. It possesses the capability to compute relevant calculations relatively faster in all the computing environments. Hence, XGBoost is widely used for its performance in modeling newer attributes and classification of labels. The application of XGBoost algorithm has gained immense momentum with its implementations in tabular and structured datasets. The evolution of XGBoost algorithm started with the decision tree based approach wherein graphical representations of possible solutions for a decision is computed depending on certain conditions. Then, an ensemble meta algorithm aggregating predictions from various decision trees based on majoritarian voting technique was created named 'bagging'. This bagging approach further evolved to construct a forest or aggregation of decision trees by randomly selecting features. The performance of the models was boosted by reducing the errors from building sequential models. As a further improvement the gradient decent algorithm was employed to reduce the errors in the sequential model. Finally XGBoost algorithm was identified as an helpful approach to optimize the gradient boosting algorithm by removing missing values, eliminating overfitting issues using parallel processing. The system optimization in XGBoost algorithm is achieved by implementing parallelization, Tree pruning and Hardware optimization as shown in Figure 1. The algorithm supports three forms of gradient boosting namely, gradient boosting machine for the learning rate; stochastic gradient boosting consisting of sub-sampling and regularized gradient boosting , which includes L1 and L2 regularizations.
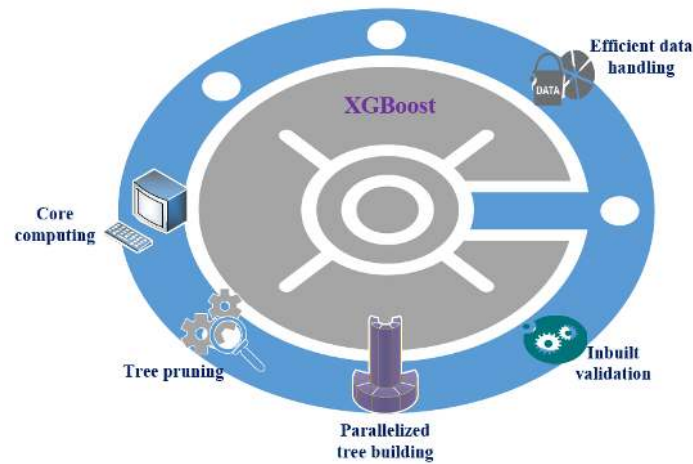
Figure 1. XGBoost.

Principal component analysis (PCA): PCA is one of the dimensional reduction techniques used in the selection and extraction of data features [27]. Feature selection is the method of transforming data into useful features by reducing size of the data. PCA reduces the variable count based on the significance using orthogonal linear combinations of the original parameters with the most significant variance.

The basic knowledge of PCA is briefly described in the following.

Assume $a_1, a_2, a_3...a_n$ are stochastic n dimensional input data records represented by a matrix A_{m*n} as shown in Equation (1).

$$A_{m*n} = \begin{bmatrix} a_{11} & ... & a_{1n} \\ a_{21} & ... & a_{2n} \\ ... & ... & ... \\ a_{m1} & ... & a_{mn} \end{bmatrix} = [a_1, a_2, ..., a_m] \tag{1}$$

1. Mean: Assume $a_1, a_2, a_3, ...a_m$ indicates the arbitrary variables for sample size m. The average of the dataset is a arbitrary parameter as shown by Equation (2).

$$\overline{A} = \frac{1}{m} \sum_{i}^{m} A_i \tag{2}$$

2. Standard deviation (S): For standard deviation calculation, the standard distance from the data set A_i must be determined at a certain point $\overline{A}$. The computation of S involves measuring distance square from all data points to the average set. The data pints are counted and partitioned to obtain a positive square root, as shown in Equation (3).

$$S = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (A_i - \overline{A})^2} \tag{3}$$

3. Covariance: Covariance configuration is very much the same as variance configuration as shown in Equation (4).

$$Cov(A, B) = \frac{\sum_{i=1}^{m} (A_i - \overline{A})(B_i - \overline{B})}{m} \tag{4}$$

4. Eigenvalues and eigenvectors of a matrix: If X is an $m*m$ matrix, then $A \neq \overline{0}$ is an eigenvector of X, where μ is a scalar for eigenvalue Y and $A \neq \overline{0}$.

5. Cumulative proportion: The cumulative proportion of sample variance explained by the first k principal components is calculated as shown in Equation (5).

$$\frac{\lambda_1 + \lambda_2 + ... + \lambda_k}{\lambda_1 + \lambda_2 + ... + \lambda_p} \tag{5}$$

where λ_k is kth eigenvalue, p the number of variables.

6. Mahalanobis distance: The Mahalanobis distance measures the distance from each point in multivariate space to the overall mean or centroid, utilizing the covariance structure of the data is shown in Equation (6).

$$Y_i = \sqrt{\left(\left(Y_i - \overline{Y}S^{-1}\right)\left(Y_i - \overline{Y}\right)\right)} \tag{6}$$

where Y_i data value vector at row i, $\overline{Y}$ mean vector, S^{-1} inverse of the covariance matrix

Firefly algorithm: Among numerous Swarm intelligent algorithms, firefly algorithm is one of modern nature-inspired algorithm developed by Yang in 2007. Based on the firefly characteristics, firefly algorithm is mainly used to solve complex problem [28]. As per the genetic nature of fireflies, any firefly can be fascinated by other firefly and they don't have discrimination with respect to sex. Due to the brightness of the fireflies, attraction among any two fireflies gets increased and similarly when the two fireflies are distant the attraction decreases. Thus the attraction among fireflies is directly proportional to the brightness based on the distance between two fireflies. This entire process is determined by an objective function $f(x)$, where $x = x_1, x_2, x_3...x_n$.

Firefly algorithm follows a different process compared to other bio-inspired algorithms. Here, at the beginning, brightness of each firefly can be measured by the objective function. Next the initial population of firefly is generated, thereby determining the light intensity of all fireflies for the generated population. The distance between two firefles is calculated based on the light intensity of two fireflies $\left(x_i, x_j\right)$. If the distance is less between the two fireflies $\left(x_i, x_j\right)$ then the nearby fireflies re grouped and the attractiveness among them is computed. Similarly, the distance is calculated for all the other fireflies by updating the values of $\left(x_i, x_j\right)$. At the end, all the fireflies are ranked and the best fireflies are selected, which are in close proximity to one another.

Firefly algorithm has a high convergence rate and it is easy to find the solution for complex problems with limited population.

4. Proposed Methodology

This section discusses about the proposed PCA-firefly based XGBoost model in detail. Figure 2 presents the proposed methodology for classification of IDS dataset. The dataset is collected from the open source Kaggle repository. Data pre-processing is an extremely crucial aspect in data analysis, which has direct effect on the accuracy of prediction. In majority of the cases, the dataset consists of missing values, missing attributes, heterogeneous and noisy data with outliers and irrelevant information included. Several techniques have been adopted like aggregation, sampling, random sampling, stratified sampling, discretization, binarization, attribute transformation and dimensionality reduction for the pre-processing of data. As part of our study a three tier approach is opted for pre-processing of the data set.

The dataset used in our study consisted of 43 attributes holding categorical and numerical data, which was not apt for training of the ML algorithm. In order to transform the data to a common format, One-Hot encoding technique was adopted, which converted categorical data to numerical values. The standard scaler technique was then applied on the dataset, which standardizes the features and transforms the mean of the distribution to 0 ensuring that most of the dataset values range between 0 and 1. The PCA algorithm is further applied on this transformed dataset for dimensionality reduction. This approach basically helps to reduce the number of random variables to precise set of principal variables thereby contributing towards accuracy in the prediction results. To optimize

more, firefly optimization algorithm selects the best attributes from this precise reduced dataset. Firefly algorithm is an approach based on the biological phenomenon pertaining to the behaviour of fireflies. A firefly generally attracts, protects its partner by flashing fluorescent lights emitted from its body. The intensity of the light depends on the distance of one fly from the other and also intensity of the light after absorption in the air. The same phenomenon is used for optimized searching for dimensionality reduction wherein the most significant attributes are considered in the machine learning model for prediction. To train this dataset, XGBoost algorithm takes an iterative boosting approach using an advanced ensemble technique for the training of the dataset. Instead of training the models in isolation, in the iterative boosting approach, the errors generated by the preceding boosting trees are corrected by the successive ones until optimized result is achieved delivering higher accuracy in prediction.

The model commences by performing One-Hot encoding [29] for transformation of the data. One-Hot encoding is a technique in machine learning pre-processing stage that help in converting categorical data into a form, which is suitable for ML algorithms. The hybrid PCA-firefly algorithm is applied on the transformed dataset for dimensionality reduction.

In machine learning models dimensionality reduction plays an extremely major role in reducing the number of redundant attributes considered, thereby reducing the time complexity by selecting the most significant attributes contributing towards improvement of prediction results. In our study, a hybrid of PCA-firefly algorithm is used to achieve dimensionality reduction. PCA basically will help to identify low dimensions for summarization of data from high dimension data. This ensures the elimination of redundant attributes from the dataset.
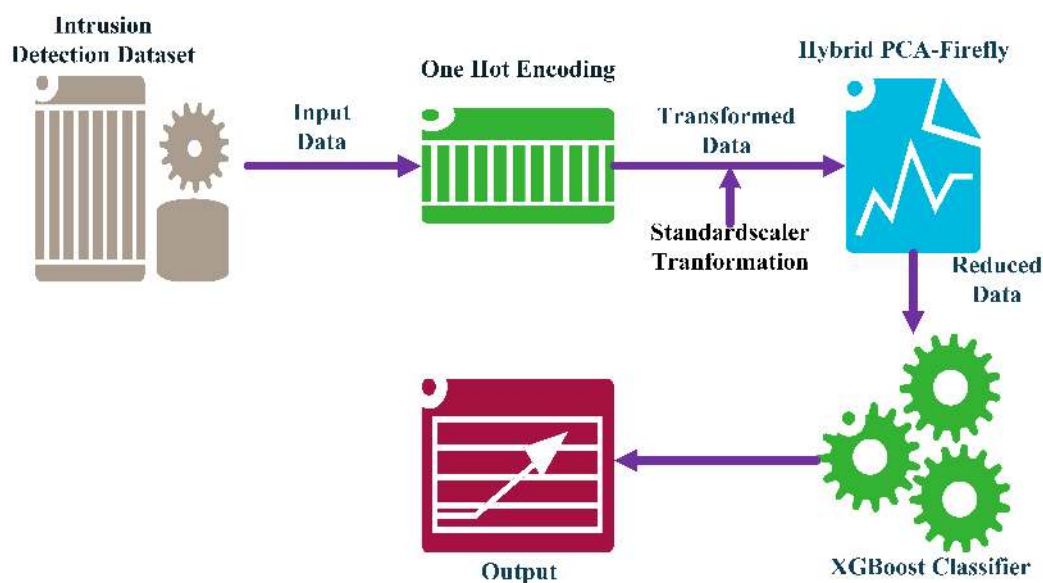

Figure 2. Proposed methodology for classification of an intrusion detection system (IDS) dataset.

To further improve on the selection of features, firefly algorithm (FA) is used in the proposed method. The implementation of FA has dual advantages. Firstly, the whole population is subdivided into groups and each group, crowd around a single mode—the local optimum. The best solution is selected from all these modes. Secondly, the subdivision of groups allow all local optima to be found simultaneously when the size of the population is higher than the number of modes. Hence the most relevant features or attributes are selected for training the ML algorithm. This concept in FA will contribute towards reduction of training time and later when machine learning algorithm is applied on the reduced data with optimized and relevant features, there will be considerable improvement in the classification results.

As mentioned earlier, the reduced dataset is further exposed to XGBoost ML algorithm for classification. Finally, the superiority of the framework is justified and established by the comparing the performance of the proposed model with traditional ML algorithms. In this work we chose XGBoost classifier as XGBoost has several advantages when compared with other ML algorithms.

Some of the advantages of XGBoost algorithm are given below:

- As XGBoost has inbuilt L1 and L2 regularization, it does not suffers from overfitting problem.
- It is much faster than gradient boosting algorithms as it utilizes parallel processing.
- It has the capability to handle missing values.

The algorithm for the proposed model is given below:

1. Intrusion detection dataset collected from Kaggle is fed to One-Hot encoding scheme to convert all the categorial data into numeric data.
2. Using the standard scaler method, all attributes of the transformed data is normalized into a range between 0 to 1.
3. Apply PCA algorithm for dimensionality reduction.
4. Apply firefly optimization algorithm to select best attributes from the reduced dataset. The firefly algorithm is illustrated below:

   (a) N no of solutions are randomly generated.
   (b) Fitness value is calculated using Equation (7).

   $$M = F_{PCA} + c + O_P \qquad (7)$$

   where M is the fitness value in the present work, F_{PCA} is features obtained by applying PCA, O_p is the objective function considered as accuracy, c is the constant ranging between $[0, 1]$
   (c) Update the solutions using the following Equation (8).

   $$F_i^{t+1} = F_i^t - \gamma_0^{xt} e^{-\beta C_a^2} \left(F_j^t - F_i^t \right) + \xi_t \psi_i^t \qquad (8)$$

   where F_i^{t+1} denotes updated ith position, F_i^t denotes current ith solution, F_j^t denotes current jth solution, which is the brighter fly, ξ_t denotes the randomization parameter, ψ_i^t is a vector of random number from Gaussion distribution at time t, γ_0^{xt}, β are the constants related to the attractiveness of the firefly.
   (d) N fitness values are generated for each of the iteration in firefly algorithm.
   (e) Using 4. (a) to 4. (d) the best attributes are selected using fitness function.
   (f) The firefly algorithm terminates once all attributes are evaluated.

5. The reduced intrusion detection dataset from the previous step is then trained using XGBoost classifier.
6. Using testing data, the performance of the model is evaluated considering accuracy, specificity and sensitivity metrics.
7. The proposed hybrid model is compared with traditional ML algorithms.

The main contributions of this work are:

- Reducing the burden and time complexity of the machine learning models.
- Improvement in the performance of ML model by choosing the best features and eliminating the features, which negatively impact the performace of the classifier by using PCA.
- Further enhancement in the performance of PCA, using properties of firefly algorithm to avoid premature convergance there by choosing the optimal features.

- Use of Google Colab, a GPU based framework offered by Google for speeding up training time of the classifier.
- Evaluation of performance of the proposed model with state-of-the-art classifiers.

5. Results and Discussion

The dataset considered for testing the proposed model is huge. Commensurate to the size of the data, RAM required to accomodate the dataset during transformation, normalization, dimensionality reduction and training of the model is also massive. Hence, the proposed work is carried out in Google Colab GPU platform. The substantial additional computational power and data handling capacity provided by the GPU platform has helped in the analysis of multiple inputs simultaneously from the dataset of such large size. The hard disk provided by Google platform is 50 GB and RAM of 25 GB. Python 3.7 was used for implementation of machine learning algorithms. The dataset used in this work is collected from Kaggle [30]. This dataset has 43 attributes and 125,973 instances. Some of the important attributes of this dataset are duration, protocol type, service, source bytes, destination bytes, wrong fragments, urgent, hot, number of failed logins, logged in, number compromised, error rate, etc. The metrics used for evaluating the model are accuracy, specificity and sensitivity.

The original dataset had 43 attributes. After One-Hot encoding technique is applied, number of attributes were enhanced to 3024. After applying PCA, number of attributes have been reduced to 2694. The attributes are further reduced to 2386 with the application of the hybrid firefly algorithm. Figure 3 shows the number of attributes for all these models.
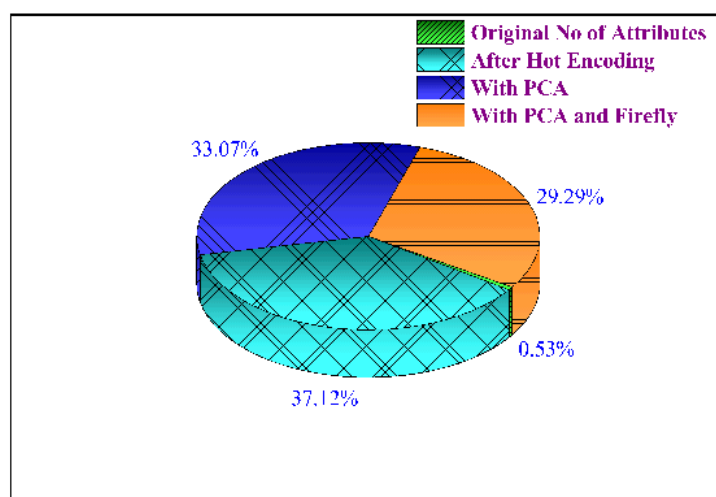

Figure 3. Number of attributes.

Figures 4–8 illustrate the performance of the KNN, naive Bayes, random forest, SVM, and XGBoost classifiers on intrusion detection dataset without dimensionality reduction, in combination with PCA and hybrid PCA-firefly algorithms.

Accuracy, sensitivity and specificity are the most important metrics popularly used to analyse the performance of the machine learning algorithms. The statistical graph in Figure 4 highlights the performance evaluation of KNN classifier based approach considering the same metrics. In case of the metrics of sensitivity, the classical KNN and KNN-PCA-firefly reveal to be equally sensitive (91.3%) with KNN-PCA (85.6%) being slightly less sensitive than the others. It is also observed that specificity values in case of KNN-PCA-firefly (99.8%), KNN-PCA (99.7%) and KNN (99.7%) are almost same with KNN-PCA-firefly being minutely higher in value. Finally it is evident from the graph that the application of KNN-PCA-firefly algorithms yields better performance than KNN (99.3%) and integration of KNN-PCA (99.2%) with higher value of accuracy (99.4%) than the others, which helps to justify the superiority of this framework.
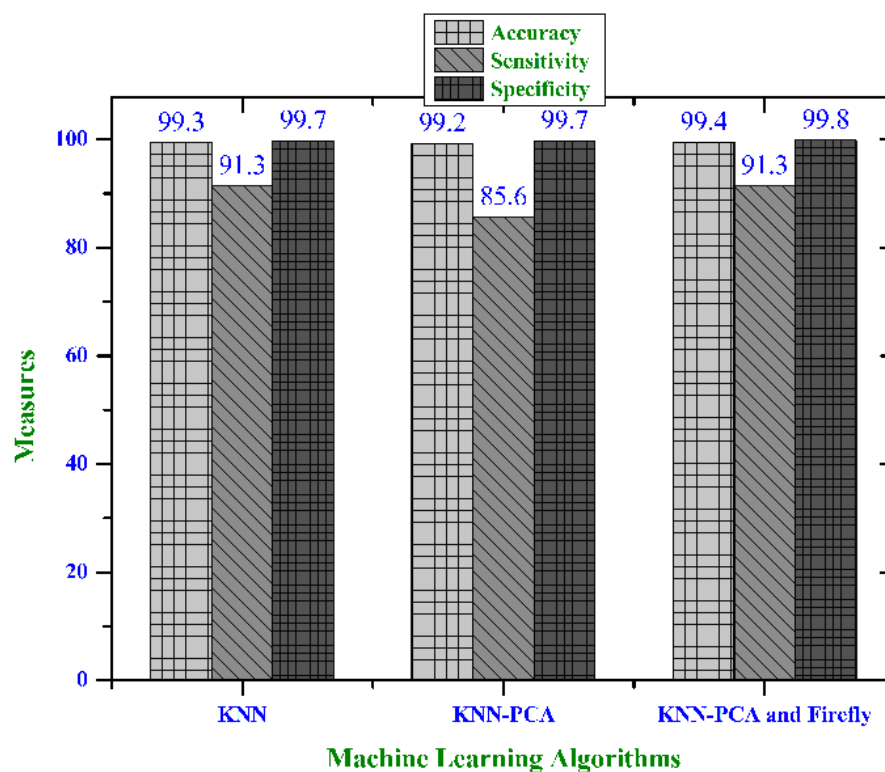
Figure 4. Performance evaluation of KNN classifier.

In case of naive Bayes based approach as shown in Figure 5, it is observed that for all the metrics (specificity, sensitivity and accuracy), the combination of Naive-Bayes-PCA-firefly algorithm yields higher value than the Naive-Bayes-PCA (accuracy—75.3%, sensitivity—68.5% and specificity—94.1%) and classical naive Bayes algorithm (accuracy—80.3%, sensitivity—73.5% and specificity—96.1%) with sensitivity value of 76.8%, specificity value of 97.2% and accuracy value of 84.2%.

In case of random forest based approach as shown in Figure 6, it is observed that for the specificity and accuracy metrics, the application of classical random forest (specificity—99.9%, accuracy—%), Random Forest-PCA (specificity—99.7%, accuracy—99.4%) and Random Forest-PCA-firefly yield almost equal values although Random Forest-PCA-firefly having slightly higher value (specificity—99.9% and accuracy—99.8%) in comparison with the other two approaches. However, in case of sensitivity, variation is observed in the values with random forest having value of 90.3%, Random Forest-PCA having value of 80.2% and Random Forest-PCA-firefly having highest value of 91.6%.

Similarly in case of support vector machine based approach as shown in Figure 7, it is observed that for the specificity and accuracy metrics, the application of classical SVM (specificity—99.6%, accuracy—97.2%), SVM-PCA (specificity—98.1%, accuracy—95.2%) and SVM-PCA-firefly yield almost equal values although SVM-PCA-firefly generates slightly higher value (specificity—99.8% and accuracy—97.5%) in comparison with the other two approaches. However, in case of sensitivity, variation is observed in the values with SVM having value of 81.2%, SVM-PCA having value of 79.3% and SVM-PCA-firefly having highest value of 84.4%.
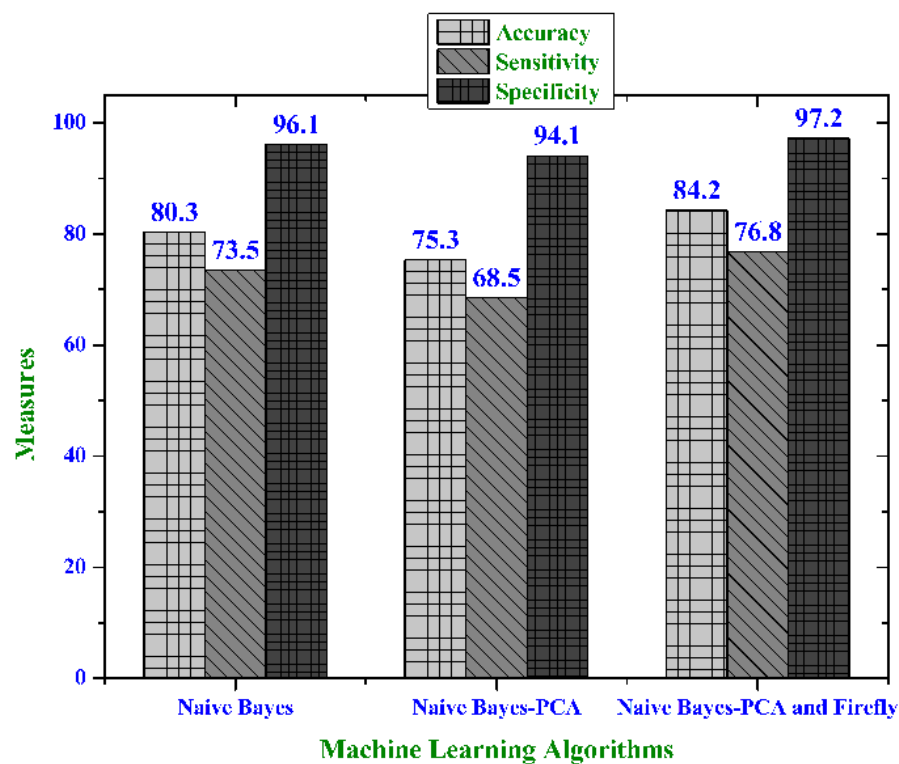
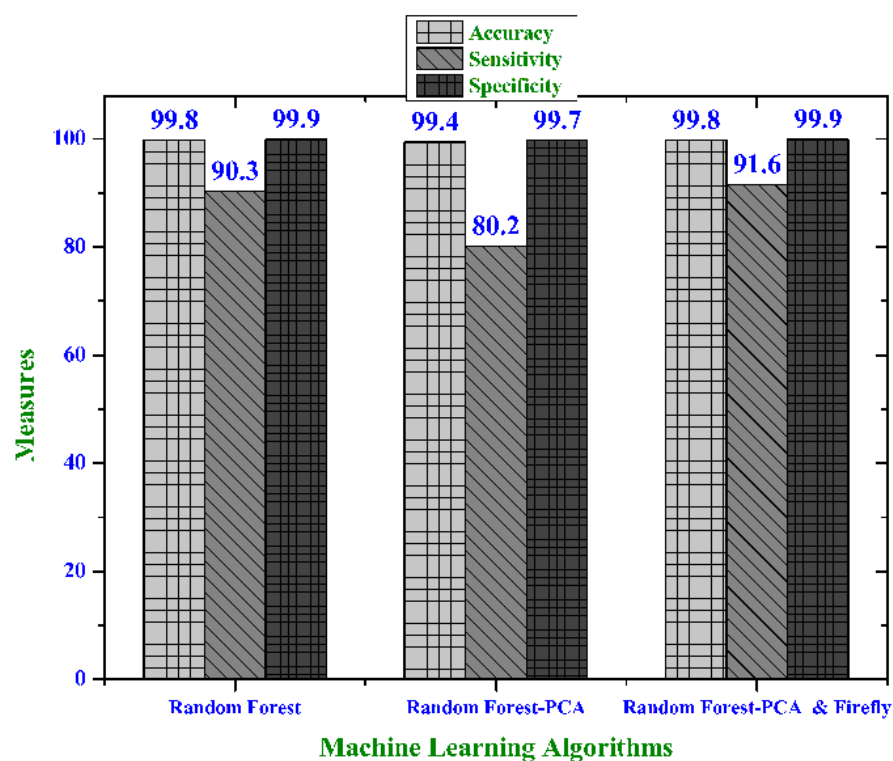**Figure 5.** Performance evaluation of naive Bayes classifier.

Figure 6. Performance evaluation of random forest classifier.

In the application of XGBoost based approach as shown in Figure 8, it is observed that for the specificity metrics, simple XGBoost (specificity—99.9%) and XGBoost-PCA-firefly (specificity—99.9%) have equal specificity in comparison to XGBoost-PCA with a slightly lower specificity value of 98.2%. Considering the accuracy metrics, all the three approaches yield almost same accuracy percentages

(99%). However, for the sensitivity measure XGBoost-PCA-firefly based algorithm clearly wins the race revealing highest value of sensitivity (93.1%) than the other two approaches (XGBoost—92.3%, XGBoost-PCA—91.8%).
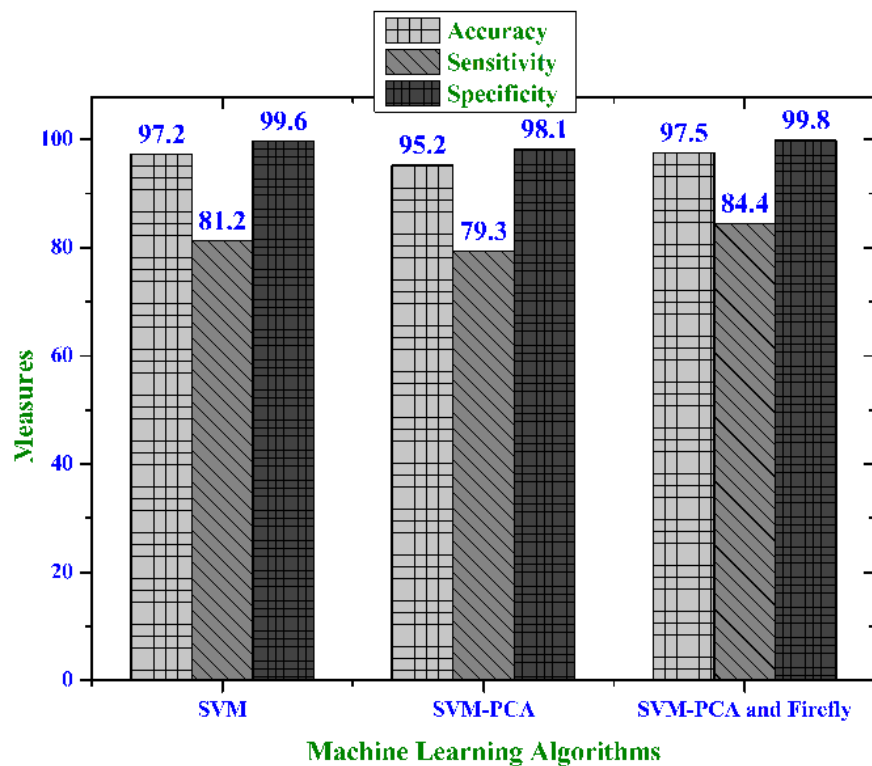

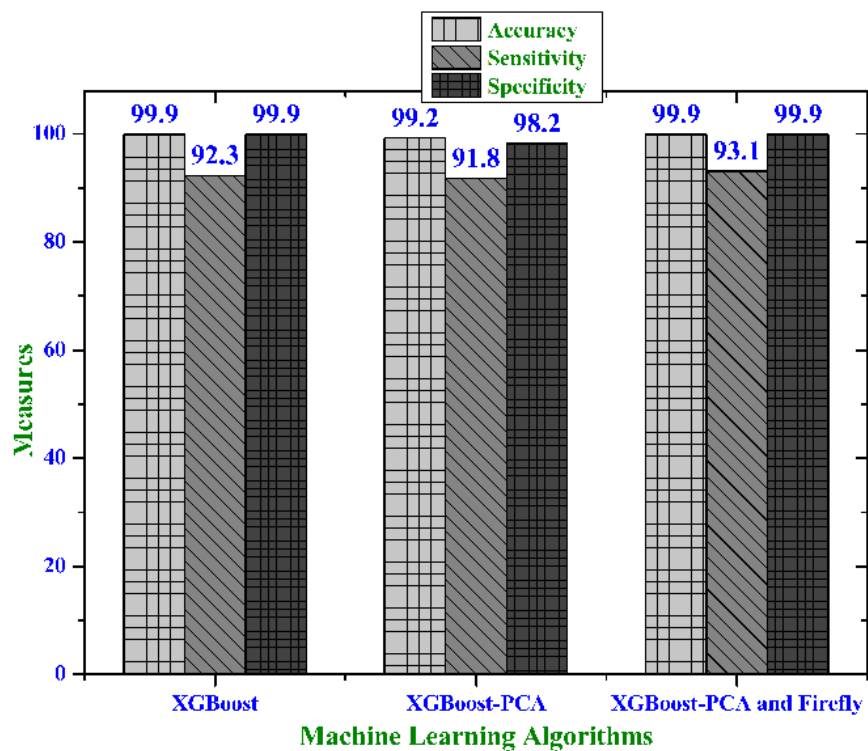**Figure 7.** Performance evaluation of SVM classifier.

Figure 8. Performance evaluation of XGBoost classifier.

The comparative analysis of all the combined approaches (KNN-PCA-firefly, naive Bayes-PCA-firefly, random forest-PCA-firefly, SVM-PCA-firefly, XGBoost-PCA-firefly) in Figure 9 highlight the fact, that the performance of XGBoost-PCA-firefly outperforms the other machine learning algorithms considering the accuracy (99.9%), sensitivity (93.1%) and specificity (99.9%) metrics. This helps to establish the fact that the predictions generated from the application of XGBoost-PCA-firefly have higher competency in decision making and hence can be relied upon as a constructive approach in machine learning.
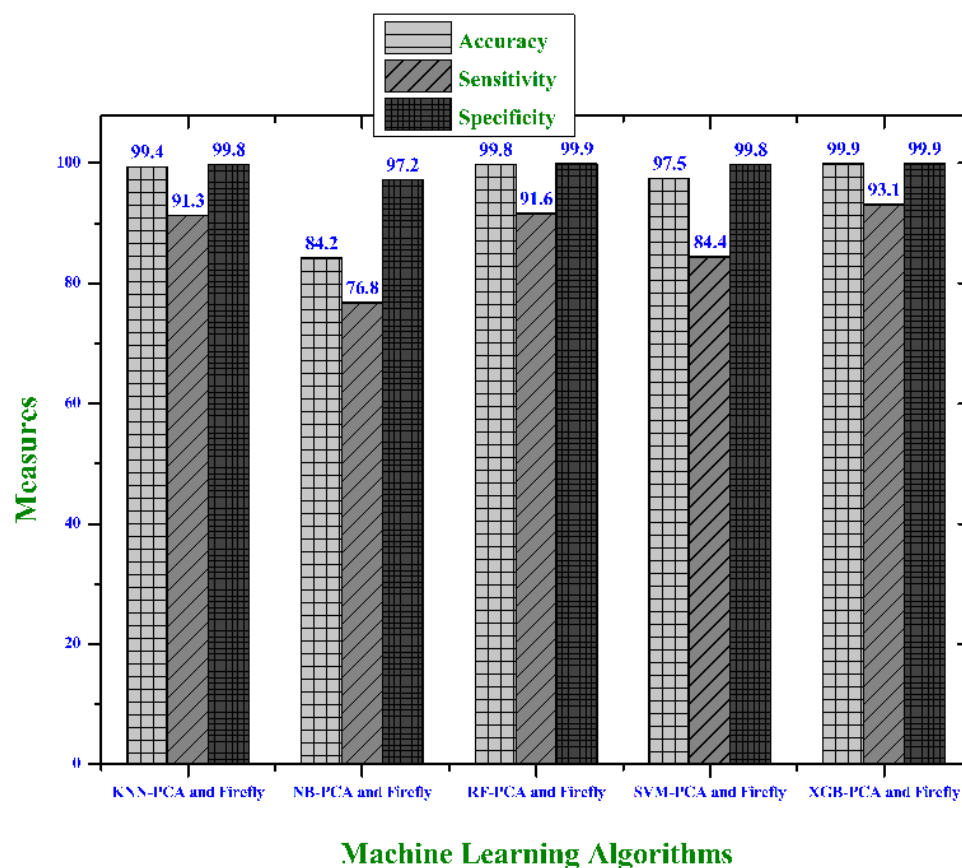

Figure 9. Comparative analysis of machine learning algorithms with proposed approach.

If the training time is keenly observed in the study, it clearly indicates the positive effect of the application of PCA and firefly on the traditional and advanced machine learning models. There is a significant reduction in time when PCA is applied on the naive Bayes, KNN, random forest, SVM and XGBoost algorithms. The training time gets further reduced when firefly algorithm is applied enhancing the performance of the machine learning model. However, if we look more precisely on the training time data of naive Bayes-PCA-firefly, KNN-PCA-firefly, random forest-PCA-firefly; the XGBoost-PCA-firefly model consumes higher training time. However, this aspect could be eliminated from a being a significant point to ponder due to its superior performance metrics namely accuracy, sensitivity and specificity. The training time for all the models considered in the experimentation are consolidated in Table 1.

From the Figures 4–8 the following points can be observed.

1. Reducing the dimensions and eliminating irrelevant attributes using PCA has improved the performance of all the classifiers.
2. Performance of the classifiers with PCA is further improved by applying firefly algorithm, as it chooses optimal features, which affect the performance of the classifier positively.

3.   Number of attributes considered for training models are drastically reduced with the application of hybrid PCA-firefly algorithm.
4.   The proposed model reduces the training time thereby reducing the burden of the classifier.
5.   The proposed hybrid PCA-firefly with XGBoost classifier model outperforms the other models considered in terms of accuracy, sensitivity and specificity with minor compromise in training time.

Table 1. Time taken for training the datasets.

NB		KNN		RF		SVM		XGB	
NB	88.7	KNN	86	RF	24	SVM	1608	XGB	280
NB+PCA	81.6	KNN+PCA	78	RF+PCA	18.7	SVM+PCA	1574.2	XGB+PCA	278.3
NB+PCA+FF	76.4	KNN+PCA+FF	71.5	RF+PCA+FF	14.6	SVM+PCA+FF	1520.3	XGB+PCA+FF	268.3

6. Conclusions and Future Work

In this research, we have proposed a hybrid principal component analysis (PCA)-Firefly based XGBoost machine learning model for the classification of IDS datasets. The dataset used in the study is a publicly available one collected from Kaggle. The framework employed starts with the One-Hot encoding approach for the transformation of the IDS dataset. The transformed data is then exposed to a hybrid PCA-firefly algorithm with the purpose of dimensionality reduction. The XGBoost algorithm is applied to this reduced dataset for classification of unanticipated cyber attacks. The results obtained from the experimental analysis suggest the proposed approach is more accurate in comparison to the traditional machine learning approaches. Based on the inferences drawn from this study, the future direction lies in implementing similar approaches towards intrusion prevention systems along with honeypots.

Author Contributions: Conceptualization, S.S., T.R.G. and U.T.; Data curation, S.S.; Formal analysis, S.B., S.R.K.S., R.K. and U.T.; Investigation, S.R.K.S., R.K., S.S. and M.A.; Methodology, T.R.G. and M.A.; Project administration, M.A.; Resources, P.K.R.M. and M.A.; Software, P.K.R.M.; Validation, S.B.; Visualization, P.K.R.M.; Writing—original draft, S.B.; Writing—review and editing, T.R.G. and U.T. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors announce that there is no discrepancy of interests concerning the publication of this research.

References

1.   Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. *arXiv* **2018**, arXiv:1806.03517.
2.   Pradhan, M.; Nayak, C.K.; Pradhan, S.K. Intrusion Detection System (IDS) and Their Types. In *Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications*; IGI Global: Hershey, PA, USA, 2020; pp. 481–497.
3.   Liu, M.; Xue, Z.; Xu, X.; Zhong, C.; Chen, J. Host-based intrusion detection system with system calls: Review and future trends. *ACM Comput. Surv. (CSUR)* **2019**, *51*, 98. [CrossRef]
4.   Deshpande, P.; Sharma, S.C.; Peddoju, S.K.; Junaid, S. HIDS: A host based intrusion detection system for cloud computing environment. *Int. J. Syst. Assur. Eng. Manag.* **2018**, *9*, 567–576. [CrossRef]
5.   Roshan, S.; Miche, Y.; Akusok, A.; Lendasse, A. Adaptive and online network intrusion detection system using clustering and extreme learning machines. *J. Frankl. Inst.* **2018**, *355*, 1752–1779. [CrossRef]
6.   Gao, X.; Shan, C.; Hu, C.; Niu, Z.; Liu, Z. An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access* **2019**, *7*, 82512–82521. [CrossRef]
7.   Ahmad, I.; Basheri, M.; Iqbal, M.J.; Rahim, A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access* **2018**, *6*, 33789–33795. [CrossRef]

8. Park, K.; Song, Y.; Cheong, Y.G. Classification of attack types for intrusion detection systems using a machine learning algorithm. In Proceedings of the 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), Bamberg, Germany, 26–29 March 2018; pp. 282–286.

9. Vinayakumar, R.; Alazab, M.; Soman, K.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]

10. Alazab, M.; Venkatraman, S.; Watters, P.; Alazab, M.; Alazab, A. Cybercrime: the case of obfuscated malware. In *Global Security, Safety and Sustainability & e-Democracy*; Springer: Thessaloniki, Greece, 2011; pp. 204–211.

11. Huda, S.; Abawajy, J.; Alazab, M.; Abdollalihian, M.; Islam, R.; Yearwood, J. Hybrids of support vector machine wrapper and filter based framework for malware detection. *Future Gener. Comput. Syst.* **2016**, *55*, 376–390. [CrossRef]

12. Khan, R.U.; Zhang, X.; Alazab, M.; Kumar, R. An Improved Convolutional Neural Network Model for Intrusion Detection in Networks. In Proceedings of the 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, Australia, 8–9 May 2019; pp. 74–77.

13. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the ICISSP, Madeira, Portugal, 22–24 January 2018; pp. 108–116.

14. Vinayakumar, R.; Alazab, M.; Soman, K.; Poornachandran, P.; Venkatraman, S. Robust Intelligent Malware Detection Using Deep Learning. *IEEE Access* **2019**, *7*, 46717–46738. [CrossRef]

15. Karatas, G.; Demir, O.; Sahingoz, O.K. Deep learning in intrusion detection systems. In Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), ANKARA, Turkey, 3–4 December 2018; pp. 113–116.

16. Othman, S.M.; Ba-Alwi, F.M.; Alsohybe, N.T.; Al-Hashida, A.Y. Intrusion detection model using machine learning algorithm on Big Data environment. *J. Big Data* **2018**, *5*, 34. [CrossRef]

17. Otoum, S.; Kantarci, B.; Mouftah, H.T. On the feasibility of deep learning in sensor network intrusion detection. *IEEE Netw. Lett.* **2019**, *1*, 68–71. [CrossRef]

18. Gadekallu, T.R.; Khare, N. Cuckoo search optimized reduction and fuzzy logic classifier for heart disease and diabetes prediction. *Int. J. Fuzzy Syst. Appl.* **2017**, *6*, 25–42. [CrossRef]

19. Reddy, G.T.; Khare, N. Hybrid firefly-bat optimized fuzzy artificial neural network based classifier for diabetes diagnosis. *Int. J. Intell. Eng. Syst.* **2017**, *10*, 18–27. [CrossRef]

20. Reddy, G.T.; Khare, N. Heart disease classification system using optimised fuzzy rule based algorithm. *Int. J. Biomed. Eng. Technol.* **2018**, *27*, 183–202. [CrossRef]

21. Reddy, G.T.; Reddy, M.P.K.; Lakshmanna, K.; Rajput, D.S.; Kaluri, R.; Srivastava, G. Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. *Evol. Intell.* **2019**, 1–12. [CrossRef]

22. Purushotham, S.; Tripathy, B. A comparative study of RIFCM with other related algorithms from their suitability in analysis of satellite images using other supporting techniques. *Kybernetes* **2014**, *43*, 53–81. [CrossRef]

23. Tripathy, B.; Ghosh, A. SDR: An algorithm for clustering categorical data using rough set theory. In Proceedings of the 2011 IEEE Recent Advances in Intelligent Computational Systems, Trivandrum, Kerala, India, 22–24 September 2011; pp. 867–872.

24. Tripathy, B. Rough sets on fuzzy approximation spaces and intuitionistic fuzzy approximation spaces. In *Rough Set Theory: A True Landmark in Data Analysis*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 3–44.

25. Chowdhary, C.L.; Acharjya, D. A hybrid scheme for breast cancer detection using intuitionistic fuzzy rough set technique. *Int. J. Healthc. Inf. Syst. Inform.* **2016**, *11*, 38–61. [CrossRef]

26. Zhao, P.; Lee, C. Assessing rear-end collision risk of cars and heavy vehicles on freeways using a surrogate safety measure. *Accid. Anal. Prev.* **2018**, *113*, 149–158. [CrossRef] [PubMed]

27. Granato, D.; Santos, J.S.; Escher, G.B.; Ferreira, B.L.; Maggio, R.M. Use of principal component analysis (PCA) and hierarchical cluster analysis (HCA) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective. *Trends Food Sci. Technol.* **2018**, *72*, 83–90. [CrossRef]

28. Moazenzadeh, R.; Mohammadi, B.; Shamshirband, S.; Chau, K.w. Coupling a firefly algorithm with support vector regression to predict evaporation in northern Iran. *Eng. Appl. Comput. Fluid Mech.* **2018**, *12*, 584–597. [CrossRef]

29. Jo, H.; Hwang, H.J.; Phan, D.; Lee, Y.; Jang, H. Endpoint Temperature Prediction model for LD Converters Using Machine-Learning Techniques. In Proceedings of the 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), Tokyo, Japan, 12–15 April 2019; pp. 22–26.
30. Kaggle. Intrusion Detection. Available online: https://www.kaggle.com/what0919/intrusion-detection (accessed on 12 December 2019).