Journal of Computer Science

Original Research Paper

Multi-Source Cyber Intrusion Detection Using Ensemble Machine Learning

[1]Taskeen Ali Khan, [2]Sara Abbas, [3]Biswaranjan Senapati, [4]Manish Raj Anand, [5]Muhammad Imran Ghafoor, [6]Satyabrata Pradhan and [7]Friban Almeida

[1]Department of Microbiology, Quaid-E-Azam University, Islamabad, Pakistan
[2]Department of Software Engineering, Islamia University of Bahawalpur, Pakistan
[3]Department of Computer Science and Data Science, Parker Hannifin Corp, USA
[4]Department of Computer Science, Medline Pharmaceuticals, Chicago, USA
[5]Department of Engineering, Pakistan Television Corporation, Lahore, Pakistan
[6]Department of Computer Science, General Motors and Automobile, Michigan, USA
[7]Department of Computer Science, Independent Researcher, New Jersey, USA

Article history
Received: 11-10-2023
Revised: 30-03-2024
Accepted: 27-05-2024

Corresponding Author:
Taskeen Ali Khan
1Department of Microbiology, Quaid-E-Azam University, Islamabad, Pakistan. Email: sirmajidsoormo@gmail.com

Abstract: The increased usage of digital technologies across businesses has led to an increase in cybercrime. It is difficult for existing intrusion detection systems to detect highly skilled hacking attempts; however, machine learning has been suggested as a way around these drawbacks. The purpose of this study is to evaluate how well various machine learning algorithms identify and stop cyberattacks in diverse network, system, and application environments. The goal of the study is to provide designers with a thorough grasp of the benefits and drawbacks of using machine learning for cyber intrusion detection. It will also assist in creating a more reliable and effective detection infrastructure. Metrics like accuracy, precision, recall, and F1-score will be used in the research to assess the models' performance. The goal is to better safeguard enterprises' networks, systems, and applications from cyberattacks by offering more precise and effective intrusion detection solutions. The objective is to determine future research areas for machine learning-based cyberattack detection methods. People all over the globe can now connect thanks to cloud computing and the Internet of Things and computer security professionals utilize standard operating procedures and proprietary software to guarantee that digital evidence is admissible in court. In digital cyber forensics, the goal of this project is to provide a revolutionary machine learning-based approach for protecting data integrity and identifying cyber threats. The best accuracy (97%), for identifying cyber hazards, will be achieved by using the Hybrid KNN-XGB, Hybrid KNN-CBC, Hybrid KNN-LGBM, Hybrid KNN-HGBC and Hybrid KNN-GBC Boosted algorithms.

Keywords: Machine Learning, Cybercrimes, Cyber Security

Introduction

The increased usage of digital technologies across businesses has led to an increase in cybercrime. Modern intrusion detection systems have trouble spotting complex hacking attempts. Machine learning has been suggested as a remedy as it makes it possible to identify new attack patterns and previously undetected attacks. In order to identify cyber-intrusions in a variety of network, system, and application situations, this project will evaluate several machine-learning techniques (Zhang *et al*., 2024). We will assess the performance of these models using deep learning, and supervised, and unsupervised learning techniques. We will investigate ensemble and hybrid models to improve the resilience and accuracy of the detection system. The goal of the research is to provide designers with a thorough grasp of the benefits and drawbacks of using machine learning for cyber intrusion detection (Naeem *et al*., 2023a). The study examines how well various machine learning algorithms identify cyber-intrusions in various systems and applications. Figure (1) shows data on cybersecurity concerns.

Fig. 1: Cyber-security concerns data

Insiders are people who have been permitted to access a company's network, system, or data; this access often leads to bad things happening (Al-Ambusaidi *et al*., 2024). Because they have authorized access to information and are aware of the vital infrastructure of the company, these assaults are hard to identify (Chawla, 2022). Early techniques for identifying insider risks relied on a particular kind of audit data, but insider threat detection has received a lot of attention lately. However, to improve the capacity to identify different kinds of assaults, current research has concentrated on studying user behaviors from many sources of audit data (Peng *et al*., 2024). Because real-world network environments don't include labeled harmful examples, unsupervised machine learning is more realistic. Nevertheless, it is still difficult to extract features using the best unsupervised machine learning model, particularly when clustering unsupervised machine learning is involved (Naeem *et al*., 2023b). The automatically learned feature representations of multi-source user activity sequences may be fed into a deep clustering network for insider threat detection, using deep neural networks (Wang *et al*., 2023). The strategy for detecting insider threats presented in this research is based on deep clustering of behavioral data from several sources (Naeem *et al*., 2023c).

The contribution of this research is to improve the precision and reliability of detection systems by combining ensemble and Hybrid models. It evaluates the models' efficacy using metrics like accuracy, precision, recall, and F1-score. The goal is to provide organizations with more accurate and efficient intrusion detection solutions to protect their networks, systems, and applications from cyber-attacks. The study also explores the success of privacy-preserving methods in the cyber-intrusion detection industry. The goal is to identify directions for further study of cyber-attack detection using machine-learning techniques. The study uses various algorithms, including Hybrid KNN-XGB, Hybrid KNN-CBC, Hybrid KNN-LGBM, Hybrid KNN-HGBC, and Hybrid KNN-GBC Boosted, to identify cyber risks with the highest accuracy (97%).

Literature Review

A multi-output decoder is designed to predict the entities of the next user behavior event that follows a user behavior event sequence and an encoder-decoder model is built to learn the feature representation of user behavior sequences using multi-source behavior audit logs. By recording the temporal correlations between user behavior events and entities of user behavior events, the encoder-decoder model may acquire initial user behavior characteristics. The majority of current techniques for detecting insider threats involve audit data to analyze user behavior, making this an important field of study Gauthama Raman *et al.* (2021). These techniques, which concentrate on identifying unusual changes at the host level, include host-based and network-based techniques. While file-based malicious insider detection approaches derive multi-dimensional feature vectors from file access pathways, keyboard, and mouse-based detection methods extract features based on click frequency and key-up and key-down durations (Soomro *et al*., 2022). Network-based techniques create user profiles based on network behavior to identify unusual application-level infractions such as sending and receiving emails and visiting websites. Nevertheless, the capacity of these techniques to identify more intricate insider threats is restricted. To enhance user behavior analysis capabilities, researchers have been experimenting with merging host-based and network-based data from multi-source audit logs. Examples include Beehive, which extracts 15-dimensional features every day, PRODIGAL, which extracts over 100-dimensional features from numerous audit logs and ensemble-based clustering algorithms for identifying abnormalities in multi-source user behavior (Naeem *et al*., 2023d). These techniques, however, depend on artificial feature engineering, which necessitates feature extraction via human labor. Low-dimensional vector representations of user behavior have been obtained via the application of embedding learning methods, however, this approach has drawbacks. Every user behavior is represented as an embedding vector in the proposed community-based anomaly detection model, which is then iteratively optimized using k-means clustering (Hossain *et al*., 2024). The temporal linkages between user activities are ignored in favor of analyzing static behavior. Conventional fossil fuel-based power generation techniques are being replaced by renewable energy sources and environmentally beneficial technology. Via reduction, reuse, and recycling, solar thermal technology has a plethora of alternatives for effective resource usage (Soomro *et al*., 2023). The viability of solar-wind hybrid systems as well as maintenance and cleaning methods for solar photovoltaic panels are being investigated by researchers. It is essential to comprehend the behavior of electricity consumers to encourage renewable energy reforms in grid firms (Hamid *et al*., 2016). By identifying consumers who are sensitive to

electricity and offering individualized services, user profile technology assists electric utility providers in lowering complaint rates and improving customer happiness (Prachi, 2017). CISCO invented fog computing, which disperses IT infrastructure, including computation, storage, and intelligence management, to data-generating devices. By putting this technology into practice, network latency may be decreased and cloud data transfer can be enhanced. Outside assaults such as U2R, PROBE, R2L, and DDoS may affect fog nodes (Buczak and Guven, 2016). Traditional network security solutions find it difficult to prevent intrusions from many sources and across domains. To identify intrusion assaults and other security breaches, intrusion detection and prevention systems, or IDPS, are necessary (Naeem *et al*., 2024a). The two main approaches to intrusion detection are signature-based and anomaly-based techniques. While anomaly-based methods create a pattern of normal behavior, signature-based techniques compare the present network activity to known dangers (Naeem *et al*., 2024b). Anomaly-based intrusion detection techniques include machine learning-based detection, data-mining-based detection, and statistics-based detection (Binbusayyis and Vaiyapuri, 2019). Intrusion Detection and Prevention Systems (IDPS) are divided into two main categories: Host-based (HIDS) and network-based (NIDS). Results from hybrid solutions, which combine the best features of both technologies, may be more successful and efficient (Geetha and Thilagam, 2021). With the Internet of Everything (IoE), cyber security is a major problem. One important intrusion prevention method is Specific Emitter Identification (SEI) technology (Awadallah Awad, 2021). For superior identification performance, this research offers a Multisource Heterogeneous Attention-based Feature Fusion Network (MHAFFN) and a Multisource Heterogeneous SEI (MH-SEI) technique (Panda *et al*., 2011). For RF fingerprinting, MHAFFN employs a multi-channel convolutional network and for automated classification, it utilizes an attention-based RFF fusion module. In an ideal setting, MHAFFN's recognition accuracy is 99.196%, yet it still has benefits in noisy surroundings (Umer *et al*., 2022). Additionally, techniques for person re-identification via domain generalization are presented in the article, including knowledge distribution augmentation and accumulation. It is essential to secure IoT networks and applications with the help of ML-enabled IoT-based IDS (Senapati and Rawal, 2023).

Materials and Methods

This section contains the methodology of the study.

Proposed Methodology

The central jail system, the central hospital database, the cyber-crime department of the banking sectors, and the cyber-crime cell all contributed data to the police department's inquiry. All of these databases were originally compiled by Pakistani intelligence agencies. First, we took the raw data and preprocessed it by eliminating the null values, standardizing the measurements, and so on. We used feature engineering (Pearson correlation) to identify the most strongly correlated feature following data cleansing and organization. Figure (2) shows the proposed methodology flow of the study.

Dataset Description

For this investigation (Police Department), information was collected from the cyber military operations Centre, the central hospital database, the central jail system, the cyber-crime department of the banking sectors, and the cyber-crime cell. Table (1) describes the dataset's features.

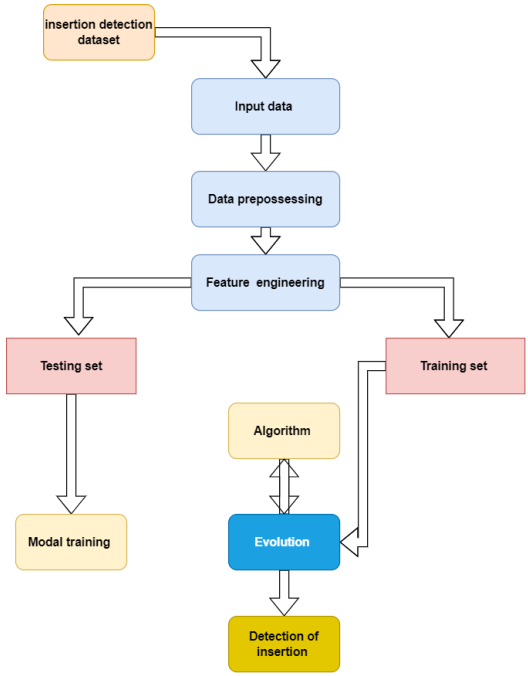

Fig. 2: Proposed flow of study

Table 1: Features description

Feature	Description	Value	Feature type
Evidence ID	Evidence saved on digital network	Any integer value	Input feature
Organization ID	Cyber military operations, centralized hospital data, central jails, cyber-crime departments of banking sectors, and cyber-crime cells (Police Departments) are all examples of organizations that use this identifier	1. Cyber military operations 2. Centralized hospital database 3. Central jails 4. Cyber-crime department of banking sectors 5. Cyber-crime cell (Police Department)	Input feature

Crime	Crime detail attempted	Name of crime (String value)	Input feature
Source Packets	Network source packets	Integer value	Input feature
Destination packets	Network destination packets	Integer value	Input feature
Cyber threat	Cyber-attack attempted on Cyber-Defense Intrusions	0 for no threat and 1 for threat or any cyber anomaly	Output feature

Data Pre-Processing

Cyber intrusion detection using several machine learning methods requires thorough data pretreatment before proceeding with the research. The primary purposes of data preprocessing include data cleaning, data transformation, and data preparation. The following procedures will be carried out to prepare the data for analysis: Scrubbing the data: Eliminating unnecessary or redundant information is the first stage in the data preparation process. This will verify that the information is trustworthy. Managing data with gaps: Taking care of data gaps is the next stage. Methods such as deleting missing rows or columns or imputed values based on the mean or median can do this.

Data Balancing

To guarantee that the data utilized in the analysis is truly representative of the population of interest, data balancing might be performed. This is crucial in classification issues where one class has a disproportionate number of observations over the other. Poor model performance may result from a biased dataset. Several methods are available to achieve a more equitable distribution of data. Over-sampling is a method that uses extra observations from the underrepresented group to create a more representative sample. The correlation matrix between the various datasets is displayed in Figs. (3-7). Figure (3) shows the dependence on bank dataset correlation. Figure (4) shows the military dataset correlation. Figure (5) shows the correlation with cybercrime cell statistics. Figure (6) shows the hospital dataset correlation. Figure (7) shows the central jail correlation dataset.

Hybrid Classification Algorithms

To boost the effectiveness of the KNN method, researchers have developed Hybrid versions of the algorithm called Hybrid K-Nearest Neighbors (KNN). The KNN algorithm is a straightforward and powerful classification method predicated on the hypothesis that cases with comparable characteristics would also have similar class labels. However, it has problems with complicated datasets and is prone to overfitting. Hybrid KNN classification methods can improve the model's performance by overcoming these restrictions brought on by the KNN algorithm alone.

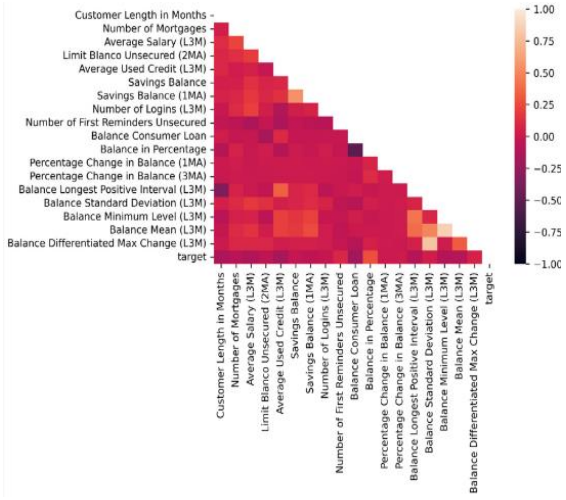

Fig. 3: Dependence on bank dataset correlation

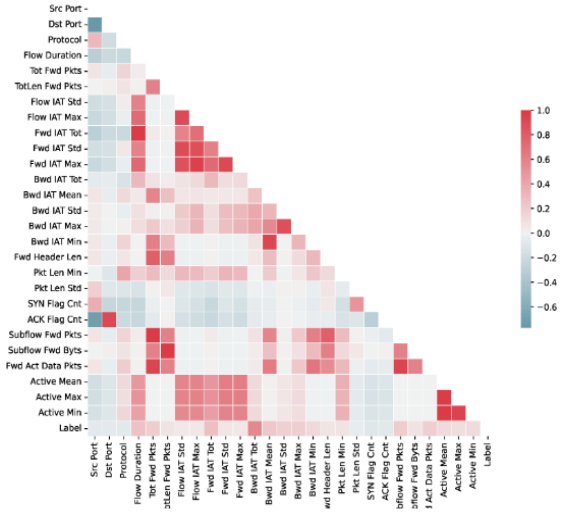

Fig. 4: Military dataset correlation

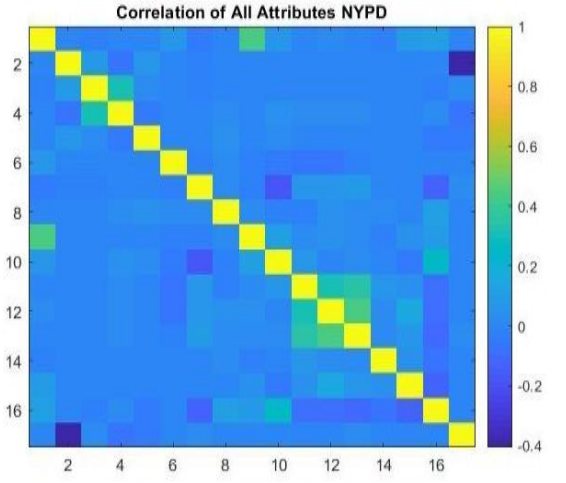

Fig. 5: Police dataset shows correlation with cybercrime cell statistics

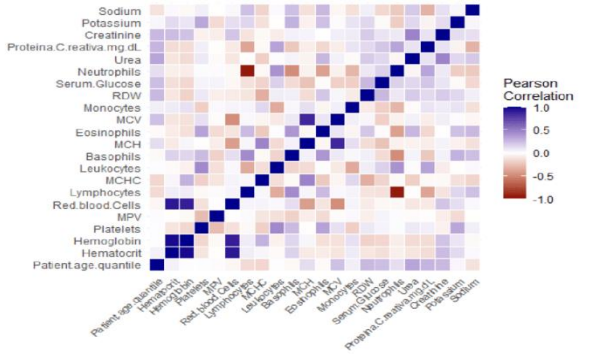

Fig. 6: Hospital dataset correlation

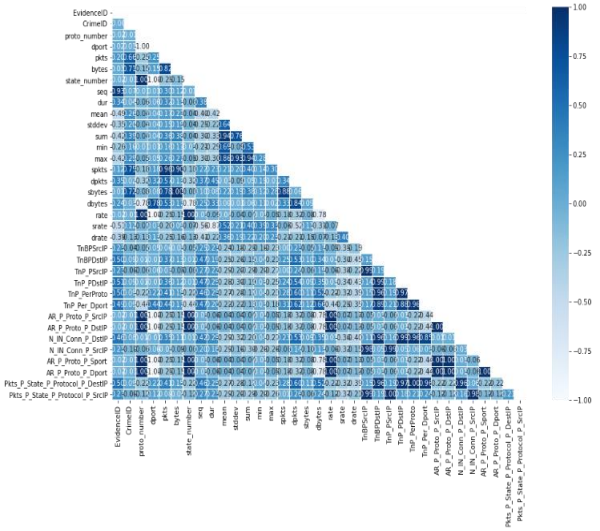

Fig. 7: Central jail correlation dataset

Hybrid ID KNN- SVM

The Hybrid KNN-SVM technique demonstrates how the K-Nearest Neighbors (KNN) algorithm can benefit from the addition of the Support Vector Machine (SVM) algorithm to the learning process. This combination approach uses the KNN algorithm to locate similar instances and the SVM algorithm to categorize the new instance according to the class labels of its neighbors. The Hybrid KNN-SVM equation is represented visually below.

K-nearest neighbor algorithm: Using a distance measure, like the Euclidean distance, find the k closest neighbors of a new instance *x*:

$$d(x,xi) = sqrt((x1 - xi1)^2 + (x2 - xi2)^2 + \ldots + (xn - xin)^2)$$

Using the SVM method, a fresh instance *x* is sorted into one of the *k* classes based on the labels of its nearest neighbors:

$$f(x) = w.x + b$$

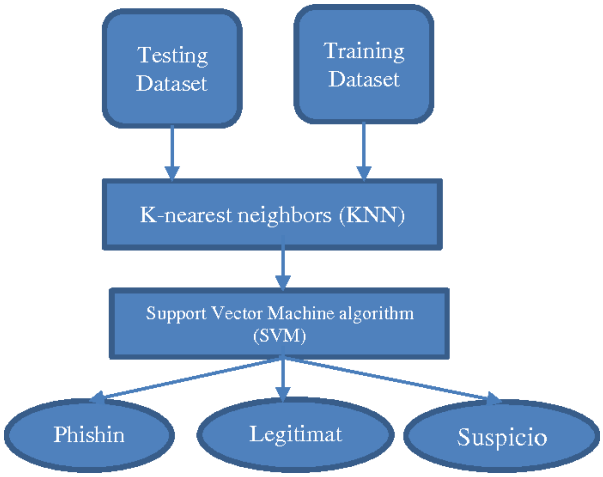

Fig. 8: Hybrid classifier (hybrid KNN-SVM classification) model process

The bias is denoted by the vector *b* and the scalar *w*. improved performance on complicated datasets is achieved by combining the KNN algorithm with the SVM algorithm to optimize the parameters and raise the model's accuracy; this is known as the Hybrid KNN-SVM technique. Figure (8) shows the Hybrid classifier (Hybrid KNN-SVM Classification) Model process.

Hybrid KNN-GA

The goal of the GA-based Hybrid KNN-GA method is to improve the efficiency of the KNN algorithm. To locate new instances that are similar to existing ones, this combined approach first employs the KNN methodology and then employs the GA algorithm to optimize the number of neighbors and distance measures utilized by KNN to optimize similarity. The Hybrid KNN-GA formula is as follows.

K-Nearest Neighbors Algorithm for Graphs Finds the k closest neighbors of a new instance x using a distance measure, like the Euclidean distance:

$$d(x,xi) = sqrt((x1 - xi1)^2 + (x2 - xi2)^2 + \ldots + (xn - xin)^2)$$

Genetic Algorithm (GA): Use a Genetic Algorithm (GA) to fine-tune the KNN algorithm. The GA is a search algorithm for optimal solutions that takes inspiration from natural selection and genetics. By incorporating the GA strategy, the Hybrid KNN-GA method improves the KNN algorithm's performance on complex datasets by helping to optimize the parameters and increase the model's accuracy. The KNN algorithm's parameters are optimized by the GA to get the best set of k nearest neighbors and distance metric for the classification task. Figure (9) shows the Model for a Hybrid Classifier (Hybrid KNN-GA).

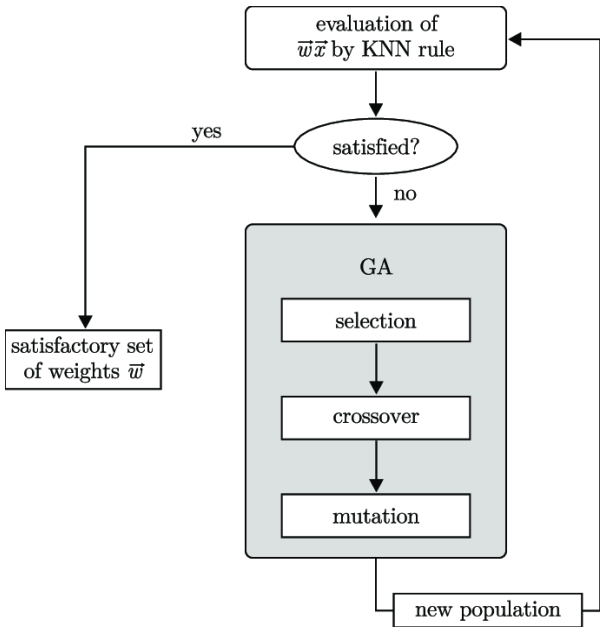

Fig. 9: Model for a Hybrid Classifier (Hybrid KNN-GA)

Hybrid KNN-BPNN

Incorporating the BPNN algorithm into the KNN algorithm results in a Hybrid known as K-Nearest Neighbors-Backpropagation Neural Network (KNN-BPNN). This Hybrid approach uses the KNN algorithm to find nearby instances with similar class labels and then the BPNN to learn the underlying patterns in the data and classify new instances accordingly. Following is a representation of the Hybrid KNN-BPNN equations: K-nearest neighbor algorithm: Using a distance measure, like the Euclidean distance, find the k closest neighbors of a new instance *x*:

$$d(x,xi) = sqrt(\,(x1 - xi1)\text{^}2 + (x2 - xi2)\text{^}2 + \ldots + (xn - xin)\text{^}2\,)$$

Backpropagation Neural Network (BPNN): A supervised learning algorithm that takes in input and employs a backpropagation algorithm to make sense of it all, using a feedforward architecture. It has many levels of connected neurons that process incoming data and output the results. Adjusting the neural network's weights based on the faults in the output is how BPNN learns the patterns in the data; the backpropagation algorithm is responsible for this. Some BPNN-related equations are as follows.

Feedforward equation:

$$y = f(w * x + b)$$

In this equation, '*x*' represents the input vector, '*w*' represents the weight matrix, '*b*' represents the bias vector, and '*f*' represents the activation function. Function error:

$$E = 1/2 * \sum(y - t)\text{^}2$$

If we define '*y*' as the projected result, '*t*' as the desired result, and '*t*' as the sum over all training instances, then we get the following.

By combining the KNN algorithm with the BPNN, which can aid in learning the underlying patterns and improving the model's accuracy, the Hybrid KNN-BPNN approach can improve the KNN algorithm's performance when dealing with complicated datasets. Classifying new instances according to the labels of their nearest neighbors, the BPNN discovers patterns in the data. Figure (10) shows a Model for a Hybrid Classifier (based on KNN and BPNN).

Hybrid KNN-DT

To boost the effectiveness of the KNN algorithm, researchers developed a method called Hybrid K-Nearest Neighbors-Decision Trees (KNN-DT). To classify a new instance according to the class labels of its nearest neighbors, this Hybrid method first employs the KNN algorithm to determine its nearest neighbors and then uses decision trees to choose the feature subset that best divides the classes. Following is a representation of the Hybrid KNN-DT equations.

K-nearest neighbor algorithm: Using a distance measure, like the Euclidean distance, find the *k* closest neighbors of a new instance *x*:

$$d(x,xi) = sqrt(\,(x1 - xi1)\text{^}2 + (x2 - xi2)\text{^}2 + \ldots + (xn - xin)\text{^}2\,)$$

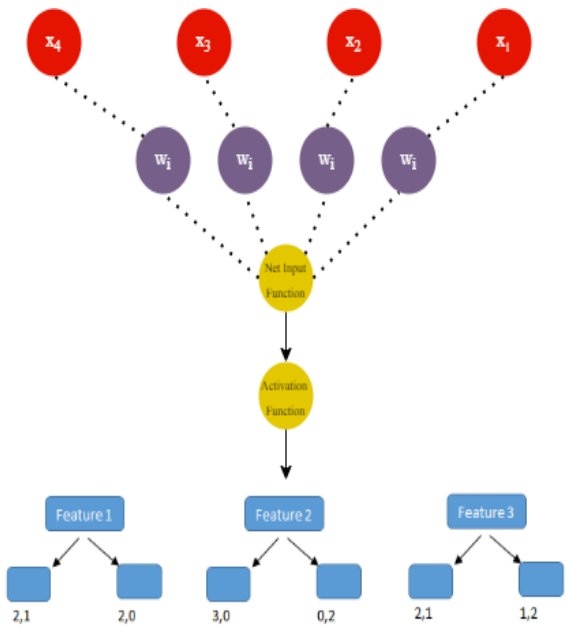

Fig 10: Model for a Hybrid Classifier (based on KNN and BPNN)

Decision Trees (DT): This is a dual-purpose tree-based model for classification and regression. It is an iterative method of dividing a dataset into smaller datasets depending on the feature values, making it an example of supervised learning. The completed decision tree consists of a set of predicating criteria applied to the incoming data. The decision tree algorithm uses these equations to identify the subset of features that most effectively discriminate between classes and to categorize new instances according to the labels of their nearest neighbors. Datasets are optimally divided using the impurity measure; decision trees are built using the CART algorithm; overfitting is minimized with pruning; and random forests are employed as an ensemble approach.

Deficiency of Gini:

$$Gini = 1 - \sum(p^2)$$

where, '*p*' is the likelihood of a class appearing in the data set. An element's Gini impurity indicates the likelihood that it would be erroneously labeled if it were labeled using the subset's label distribution.

The classification and regression tree:

$$CART = min(Cost(left) + Cost(right))$$

where, "*Cost*" is some measure of impurity, like Gini impurity or entropy, and "*left*" and "*right*" refer to the two offspring nodes. The CART algorithm is used to build the tree of judgment. To find the optimal data partition, it employs the impurity measure.

Pruning is a method for reducing overfitting in which ineffective nodes of the decision tree are removed. To prune a tree, apply the following equation:

$$Alpha = (1 - (validation\_error/training\_error)) * 100$$

In which the validation set error (*'validation_error'*) and the training set error (*'training_error'*) are replaced by their respective values. Each node's alpha is determined; if it falls below a certain level, the node in question is eliminated. Random forest is an ensemble technique that makes predictions by combining the outputs of several individual decision trees. The equation for determining the random forest's forecast is as follows:

$$Y = \left(\frac{1}{n}\right) * \sum(Yi)$$

where, '*n*' is the number of decision trees in the random forest and '*Yi*' is the prediction of each decision tree. Figure (11) shows a Model for a Hybrid Classifier (Hybrid KNN-DT).

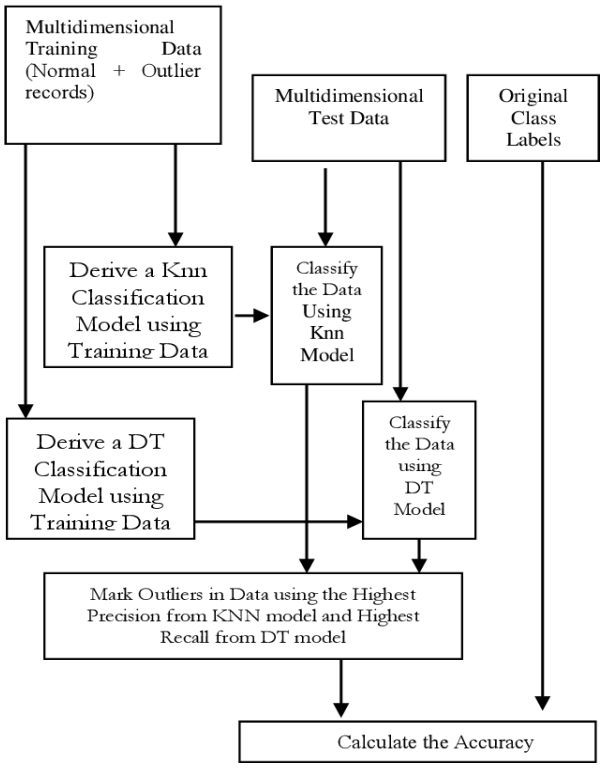

Fig. 11: Model for a Hybrid classifier (Hybrid KNN-DT)

Hybrid KNN-PCA

The KNN algorithm is enhanced by the Principal Component Analysis (PCA) algorithm in the Principal Component Analysis (Hybrid KNN-PCA) classification algorithm. New instances are classified using nearest neighbors in the reduced feature space and the PCA technique is used to reduce the dimensionality of the data and extract the most relevant characteristics. Following are the foundational building blocks of the Hybrid KNN-PCA algorithm: The PCA approach is used first to transform the data into a new set of uncorrelated variables called principal components to minimize the dimensionality of the data. The covariance matrix's eigenvectors are used to build a new set of features for the data. After sorting the eigenvectors from highest to lowest eigenvalue, the first *k* is chosen as the cornerstone of the updated collection of features. After the feature set has been narrowed down, the KNN technique is used to classify fresh instances based on their nearest neighbors. The KNN algorithm is simple since it only needs to know how far apart a new instance is from the k nearest training instances to assign it a label. The KNN-PCA model combination can be described by the following equation: Labels *Y* will be generated using data set *X*, hence we will apply the Hybrid KNN-PCA formula: *KNN*(*PCA*(*X*), *Y*)).

To classify new instances, we first run the PCA algorithm on the input dataset *X* and return the principal components; then, we run the KNN algorithm on the

principal components using the output labels *Y*; finally, we employ the PCA and KNN functions in conjunction with one another to classify the new instances. The Hybrid KNN-PCA method improves the performance of the KNN algorithm, leading to reduced computational costs and improved model accuracy, by reducing the dimensionality of the data and extracting the most significant features. Figure (12) shows a Model for a Hybrid Classifier (Hybrid ID KNN-PCA).

Hybrid KNN-Fuzzy

To enhance the performance of the K-Nearest Neighbors (KNN) algorithm, the Hybrid KNN-Fuzzy classification system incorporates Fuzzy logic into the KNN algorithm. Fuzzy logic is employed to deal with ambiguity and imprecision in the data, hence strengthening the KNN algorithm's capacity to classify cases. Here are the fundamental operations of the Hybrid KNN-Fuzzy algorithm: First, the data must be fuzzified using fuzzy logic, with the crisp values first being represented as fuzzy sets. To do this, membership functions are defined to convert the discrete data values to a fuzzy interval between 0 and 1. To categorize using KNN: The KNN technique is then used on the fuzzy data to categorize new cases using nearest neighbors, following the fuzzification stage. The KNN algorithm is a straightforward one, as it uses only the distance between a new instance and the k nearest training instances to determine its classification. The final stage is fuzzy fiction, wherein the fuzzy values are converted back to crisp values that were originally input to the KNN algorithm. Defuzzi fiction techniques such as the center of gravity, mean of maximum, and centroid are used for this step. Following are some equations that can be used to illustrate the Hybrid KNN-Fuzzy algorithm.

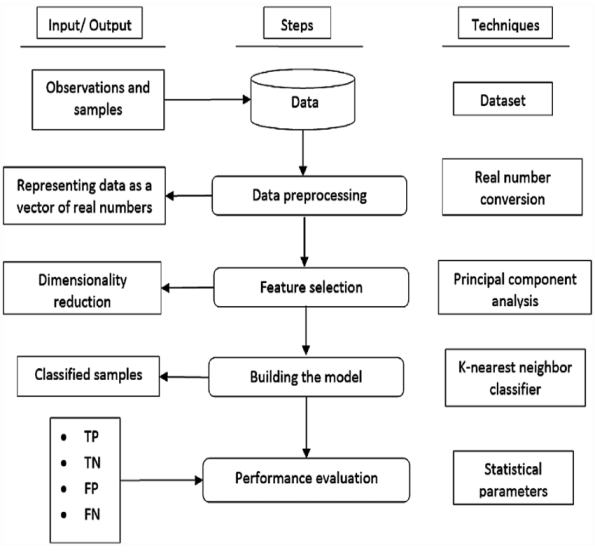

Fig. 12: Model for a Hybrid Classifier (Hybrid ID KNN-PCA)

Let *X* be the dataset to be fuzzified, *xi* the i^{th} instance of the dataset, and *mi*(*xi*) the membership function of *xi* in the fuzzy set *A*:

$$mi(xi) = \{1, if\ xi\ belongs\ to\ A\ 0, otherwise\}$$

KNN Classification

The distance between two labels, *xi*, and *yi*, is denoted by *d*(*xi*, *yi*), where *Y* is the set of final labels and is the i^{th} label in the dataset:

$$yi = KNN(X, mi(xi), k)$$

where, *k* is the number of nearest neighbors.

Defuzzi Fiction

Let *Z* be the crisp output, *zi* be the i^{th} output and *COG H*(*mi*(*xi*)) be the center of gravity defuzzification method:

$$zi = COG(mi(xi))$$

Fuzzy output is mapped to crisp output using the center of gravity defuzzification method, which is abbreviated as COG.

By using fuzzy logic to deal with ambiguity and vagueness in the data, the Hybrid KNN-Fuzzy method can boost the performance of the KNN algorithm. It makes the KNN algorithm more resilient when dealing with the classification of instances and it can process data that is ambiguous or vague. Figure (13) shows the hybrid KNN-fuzzy classification model is a hybrid classifier.

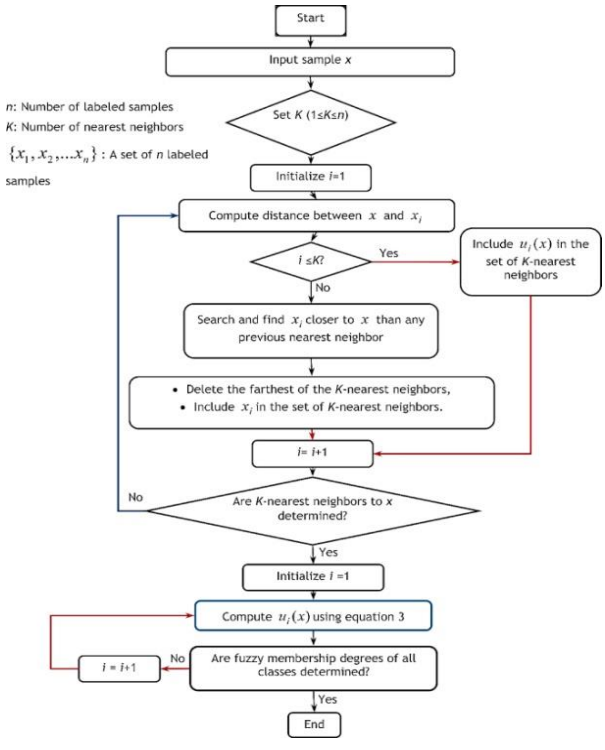

Fig. 13: The hybrid KNN-fuzzy classification model is a hybrid classifier

Table 2: Description of metrics

Metric	Description
Accuracy	$\text{Accuracy}=\frac{TP}{(TP+TN)}*100$
Confusion matrix	$\text{Precision}=\frac{\in TP}{\in TP+FP}$
	$\text{Recall}=\frac{\in TP}{\in TP+FN}$
	$\text{Accuracy}=\frac{\in TP+TN}{\in TP+FP+FN+TN}$

Performance Parameters

The precision of the system was evaluated with the help of the F1-score and other indicators. Based on the outcomes of the confusion matrix, the categorization and misclassification provisions have been given descriptive names. Data used for the analysis are presented. Table (2) shows a description of the metrics.

Results and Discussion

Hybrid KNN-XGB is a Hybrid Model

The Hybrid KNN-XGB technique uses the XG Boost algorithm to pick the most important characteristics from the dataset, hence enhancing the KNN algorithm's performance. As a result, the KNN method can use a more condensed collection of features, which can both lessen the burden on the computer and boost the reliability of the predicted results. This modal has an accuracy of 65.5-95%. Figure (14) shows the confusion matrix of the Hybrid KNN-XGB Classification Model.

Hybrid Model Hybrid KNN-GBC

By employing the GBC algorithm to pick the most pertinent characteristics from the dataset, the Hybrid KNN-GBC strategy can boost the KNN algorithm's performance. As a result, the KNN method can use a more condensed collection of features, which can both lessen the burden on the computer and boost the reliability of the predicted results.

The confusion matrix produced by the Hybrid KNN-GBC classification model is displayed in Fig. (15).

Model Hybrid KNN-ABC, a Hybrid

By employing the ABC algorithm to pick the most pertinent characteristics from the dataset, the Hybrid KNN-ABC algorithm can boost the KNN algorithm's performance. As a result, the KNN method can use a more condensed collection of features, which can both lessen the burden on the computer and boost the reliability of the predicted results. Figure (16) depicts the ambiguity matrix of Hybrid KNN-ABC.

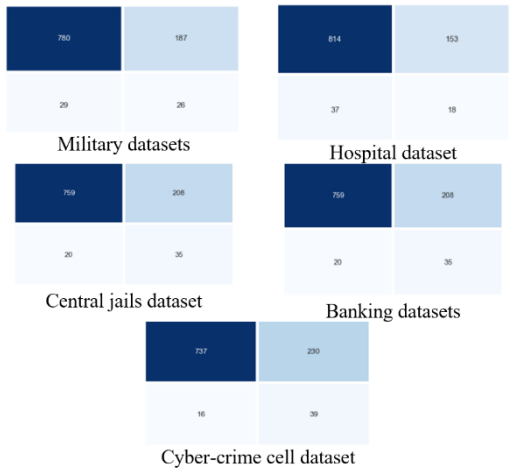

Fig. 14: The confusion matrix of the Hybrid KNN-XGB Classification model

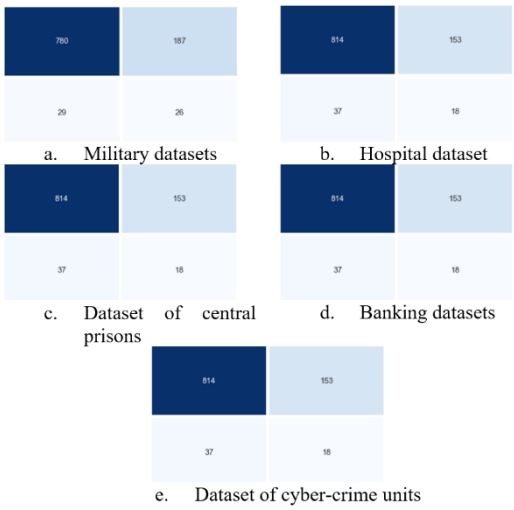

Fig. 15: Hybrid KNN-GBC Classification Matrix for Confusion

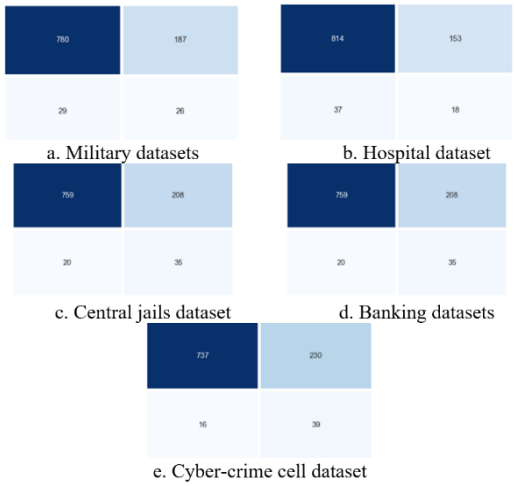

Fig. 16: Confusion Matrix of Hybrid KNN-ABC classification model

Hybrid Model Hybrid KNN-CBC

By fusing the KNN method with the Bagging Classifier (CBC) technique, we get the Hybrid K-Nearest Neighbors and Bagging Classifier (KNN-CBC). As an ensemble approach that constructs many models (decision trees) by bootstrapping the training data and averaging the predictions, CBC is a potent machine learning algorithm that can handle big datasets and perform well in both regression and classification problems. The greatest accuracy rating for this model is 97%. The Hybrid KNN-CBC Classification Model's confusion matrix is displayed in Fig. (17).

Hybrid Model Hybrid KNN- LGB

By employing the LGB algorithm to select the most pertinent characteristics from the dataset, the Hybrid KNN-LGB algorithm can enhance the performance of the KNN algorithm. As a result, the KNN approach can use a more condensed collection of features, which may enhance model accuracy while decreasing computing costs. Figure (18) presents the Hybrid KNN-LGB Classification Model's confusion matrix.

Hybrid Model Hybrid KNN- HBC

By employing the HBC algorithm to pick the most pertinent characteristics from the dataset, the Hybrid KNN-HBC algorithm can boost the KNN algorithm's performance. As a result, the KNN method can use a more condensed collection of features, which can both lessen the burden on the computer and boost the reliability of the predicted results. Accuracy was enhanced by using the Hybrid KNN model and the Histogram Gradient Boosting Classifier. When the Hybrid KNN probability function reaches HGBC, we'll examine it to determine whether any classes have been changed.

The Hybrid KNN-HGBC Confusion Matrix is displayed in Fig. (19).

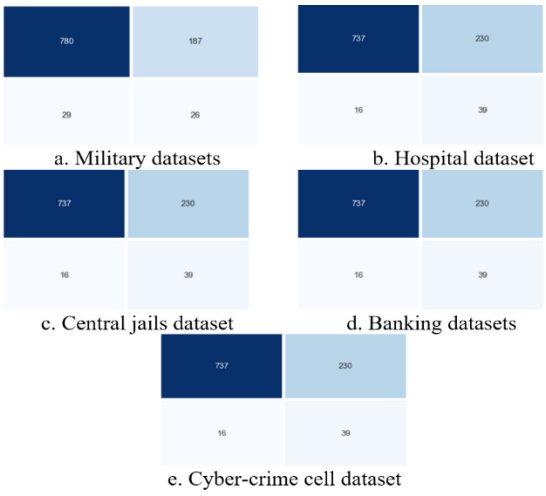

Fig. 17: The confusion matrix of the Hybrid KNN-CBC classification model

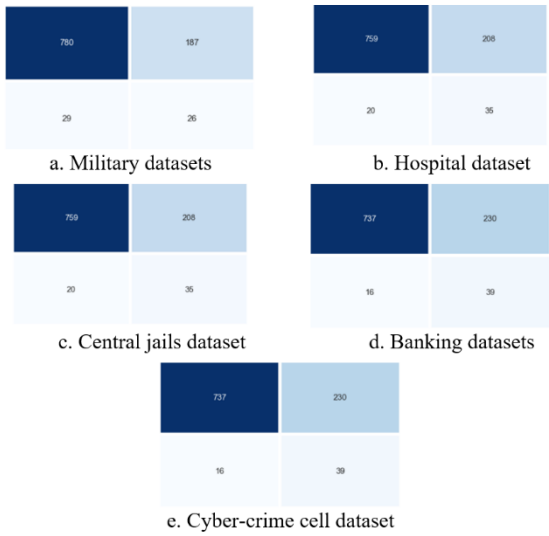

Fig. 18: The confusion matrix of the Hybrid KNN-LBC classification model

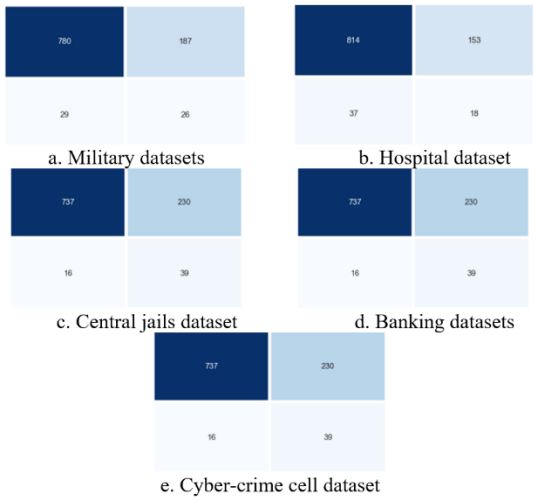

Fig. 19: The confusion matrix of the Hybrid KNN-HGBC classification model

Discussion

See how the various models compare in terms of precision in the table below. High levels of accuracy were achieved by both Hybrid KNN-XGB and Hybrid KNN-GBC, with Hybrid KNN-XGB achieving a whopping 95.5% accuracy. With the help of Hybrid KNN-ABC, its precision increased to 89.33%. The highest accuracy of the approaches we tried was achieved by Hybrid KNN-CBC (97.00%). In terms of accuracy, both Hybrid KNN-LGBM and Hybrid KNN-HGBC scored a respectable 91.37%. Figure (20) shows a Comparative Analysis for the detection of Cyber Threats to other Hybrid KNN Boosted Algorithms on this dataset, the Hybrid KNN-CBC performed the best. Figure (20) presents a comparative analysis of the detection of cyber threats.

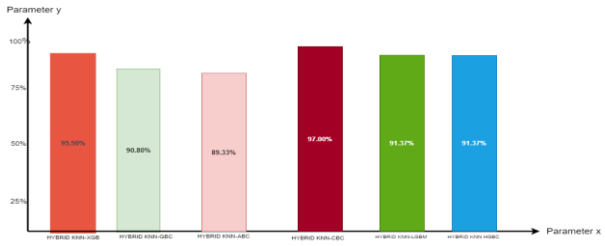

Fig. 20: Comparative analysis for the detection of cyber threat

Conclusion

The rise in digital technologies has led to a surge in cybercrime, making it challenging for existing intrusion detection systems to detect skilled hacking attempts. The intrusion detection protection has limitations with their cybercrime; however, this limitation is resolved by combining it with the hybrid machine learning method. This study aims to evaluate the effectiveness of various machine learning algorithms in identifying and stopping cyberattacks in diverse network, system, and application environments. The goal is to provide enterprises with more precise and effective intrusion detection solutions to protect their networks, systems, and applications from cyberattacks. The study also explores the success of privacy-preserving methods in the cyber-intrusion detection industry and identifies directions for further study of cyber-attack detection using machine learning techniques. The proliferation of cyberattacks and IoT dangers highlights the growing importance of data security concerns among Internet users. The study's overarching goal is to develop a novel Machine Learning-based strategy for preserving data integrity and detecting cyber threats in digital cyber forensics. Various algorithms, including Hybrid KNN-XGB, Hybrid KNN-CBC, Hybrid KNN-LGBM, Hybrid KNN-HGBC, and Hybrid KNN-GBC Boosted, are used to identify cyber risks with the highest accuracy (97%). High levels of accuracy were achieved by both Hybrid KNN-XGB and Hybrid KNN-GBC, with Hybrid KNN-XGB achieving a whopping 95.5% accuracy. Hybrid KNN-ABC increased its precision to 89.33%, while Hybrid KNN-CBC achieved the highest accuracy (97.00%). Future work will concentrate on creating a system that automatically determines the ideal number of clusters depending on the properties of the dataset being used.

Acknowledgment

We thank Taskeen Ali Khan (Quaid-E-Azam University) for writing the research article evaluation, work in Biosynthesis of silver nanoparticles from novel Bischofia javanica plant loaded chitosan hydrogel, paper methodology, and paper conceptualisation. Biswaranjan Senapati (Parker Hannifin Corp) provided software assistance, while Sara Abbas (Islamia University of Bahawalpur) re-viewed and modified the paper in response to journal reviews and simulation work. Muhammad Imran Ghafoor (Pakistan Television Corporation,) assisted with the investigation and methodology of the work. Pakistan and the United States Transportation System contributed to this study.

Funding Information

This research received no external funding.

Author's Contributions

Taskeen Ali Khan: Conceptualization, methodology, software development, field study, review and editing.

Sara Abbas: Review and edited, visualization, investigation.

Biswaranjan Senapati: Data curation, original draft preparation, software development, field study.

Manish Raj Anand: Software development, review and edited.

Muhammad Imran Ghafoor: Investigation, methodology.

Satyabrata Pradhan: Methodology, software development.

Friban Almeida: Data curation, original draft preparation.

All authors have read and agreed to the published version of the manuscript.

Ethics

This article does not contain any studies with animals performed by any of the authors.

Data Availability Statement

The authors declare that all data supporting this study's findings are available within the article.

Conflicts of Interest

The author declares no potential conflict of interest.

Reference

Al-Ambusaidi, M., Yinjun, Z., Muhammad, Y., & Yahya, A. (2024). ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications. *Soft Computing*, *28*(2), 1765–1784. https://doi.org/10.1007/s00500-023-09452-7

Awadallah Awad, N. (2021). Enhancing Network Intrusion Detection Model Using Machine Learning Algorithms. *Computers, Materials & Continua*, *67*(1), 979–990. https://doi.org/10.32604/cmc.2021.014307

Binbusayyis, A., & Vaiyapuri, T. (2019). Identifying and Benchmarking Key Features for Cyber Intrusion Detection: An Ensemble Approach. *IEEE Access*, *7*, 106495–106513. https://doi.org/10.1109/access.2019.2929487

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, *18*(2), 1153–1176. https://doi.org/10.1109/comst.2015.2494502

Chawla, A. (2022). Phishing website analysis and detection using Machine Learning. *International Journal of Intelligent Systems and Applications in Engineering*, *10*(1), 10–16. https://doi.org/10.18201/ijisae.2022.262

Gauthama Raman, M. R., Chuadhry Mujeeb, A., & Mathur, A. (2021). Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation. *Cybersecurity*, *4*(1), 27. https://doi.org/10.1186/s42400-021-00095-5

Geetha, R., & Thilagam, T. (2021). A Review on the Effectiveness of Machine Learning and Deep Learning Algorithms for Cyber Security. *Archives of Computational Methods in Engineering*, *28*(4), 2861–2879. https://doi.org/10.1007/s11831-020-09478-2

Hamid, Y., Sugumaran, M., & Journaux, L. (2016). Machine Learning Techniques for Intrusion Detection: A Comparative Analysis. *Proceedings of the International Conference on Informatics and Analytics*, 1–6. https://doi.org/10.1145/2980258.2980378

Hossain, Z., Rahman Sourov, Md. M., Khan, M., & Rahman, P. (2021). Network Intrusion Detection using Machine Learning Approaches. *2021 50th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 303–307. https://doi.org/10.1109/i-smac52330.2021.9640949

Naeem, A. B., Senapati, B., Bhuva, D., Zaidi, A., Bhuva, A., Sudman, Md. S. I., & Ahmed, A. E. M. (2024b). Heart Disease Detection Using Feature Extraction and Artificial Neural Networks: A Sensor-Based Approach. *IEEE Access*, *12*, 37349–37362. https://doi.org/10.1109/access.2024.3373646

Naeem, A. B., Senapati, B., Chauhan, A. S., Makhija, M., Singh, A., Gupta, M., Tiwari, P. K., & Abdel-Rehim, W. M. F. (2023b). Hypothyroidism Disease Diagnosis by Using Machine Learning Algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(3), 368–373.

Naeem, A. B., Senapati, B., Islam Sudman, Md. S., Bashir, K., & Ahmed, A. E. M. (2023a). Intelligent Road Management System for Autonomous, Non-Autonomous and VIP Vehicles. *World Electric Vehicle Journal*, *14*(9), 238. https://doi.org/10.3390/wevj14090238

Naeem, A. B., Senapati, B., Mahadin, G. A., Ghulaxe, V., Almeida, F., Sudman, S. I., & Ghafoor, M. I. (2024a). Determine the Prevalence of Hepatitis B and C During Pregnancy by Using Machine Learning Algorithm. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(13s), 744–751.

Naeem, A. B., Soomro, A. M., Bhuva, A., Bashir, K., Bhuva, D., Maaliw, R. R., & Abdel-Rehim, W. M. F. (2023d). Intelligent Four-Way Crossroad Safety Management for Autonomous, Non-Autonomous and VIP Vehicles. *2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICES&T)*, 1–6. https://doi.org/10.1109/icest56843.2023.10138829

Naeem, A. B., Soomro, A. M., Saim, H. M., & Malik, H. (2023c). Smart road management system for prioritized autonomous vehicles under vehicle-to-everything (V2X) communication. *Multimedia Tools and Applications*, *83*(14), 41637–41654. https://doi.org/10.1007/s11042-023-16950-1

Panda, M., Abraham, A., Das, S., & Patra, M. R. (2011). Network intrusion detection system: A machine learning approach. *Intelligent Decision Technologies*, *5*(4), 347–356. https://doi.org/10.3233/idt-2011-0117

Peng, W., Chen, H., Li, Y., & Sun, J. (2024). Multi-source domain generalization peron re-identification with knowledge accumulation and distribution enhancement. *Applied Intelligence*, *54*(2), 1818–1830. https://doi.org/10.1007/s10489-024-05266-8

Prachi. (2017). Network Intrusion Detection Using Machine-Learning Techniques. *International Journal of Data Mining and Emerging Technologies*, *7*(1), 23–32. https://doi.org/10.5958/2249-3220.2017.00004.0

Senapati, B., & Rawal, B. S. (2023). Adopting a Deep Learning Split-Protocol Based Predictive Maintenance Management System for Industrial Manufacturing Operations. *Big Data Intelligence and Computing*, 22–39. https://doi.org/10.1007/978-981-99-2233-8_2

Soomro, A. M., Naeem, A. B., Senapati, B., Bashir, K., Pradhan, S., Maaliw, R. R., & Sakr, H. A. (2023). Constructor Development: Predicting Object Communication Errors. *2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICES&T)*, 1–7. https://doi.org/10.1109/icest56843.2023.10138846

Soomro, A. M., Naeem, A. B., Shahzad, K., Madni, A. M., Del Mundo, A. D., Sajid, M., & Baloch, M. A. (2022). Forecasting Cotton Whitefly Population Using Deep Learning. *Journal of Computing & Biomedical Informatics*, *4*(01), 64–76. https://doi.org/10.56979/401/2022/67

Umer, M., Sadiq, S., Karamti, H., Alhebshi, R. M., Alnowaiser, K., Eshmawi, A. A., Song, H., & Ashraf, I. (2022). Deep Learning-Based Intrusion Detection Methods in Cyber-Physical Systems: Challenges and Future Trends. *Electronics*, *11*(20), 3326. https://doi.org/10.3390/electronics11203326

Wu, Y.; Wang, Q.; Guo, N.; Tian, Y.; Li, F.; Su, X. Efficient Multi-Source Self-Attention Data Fusion for FDIA Detection in Smart Grid. Symmetry 2023, 15, 1019. https://doi.org/10.3390/sym15051019

Wang, J., Sun, Q., & Zhou, C. (2023). Insider Threat Detection Based on Deep Clustering of Multi-Source Behavioral Events. *Applied Sciences*, *13*(24), 13021. https://doi.org/10.3390/app132413021

Zhang, Y., Zhang, Q., Zhao, H., Lin, Y., Gui, G., & Sari, H. (2024). Multisource Heterogeneous Specific Emitter Identification Using Attention Mechanism-Based RFF Fusion Method. *IEEE Transactions on Information Forensics and Security*, *19*, 2639–2650. https://doi.org/10.1109/tifs.2024.3353594