International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 1-7, February-2024

https://doi.org/10.47001/IRJIET/2024.802001

Implement an Intrusion Detection System Utilizing Machine Learning and Principal Component Analysis

[1]Rula Abdulwahid Mohammed, [2]Youssef A. Bazzi

[1]Faculty of Engineering, Islamic University of Lebanon, Wardanieh, Lebanon

[2]Department of Electrical and Computer Engineering, Lebanese University, Beirut, Lebanon

***Abstract* - The ever-evolving cybersecurity sector requires robust intrusion detection systems (IDS). Traditional rules-based measures are no longer sufficient due to the complexity of cyber threats, requiring new approaches. This study presents the architecture of an intrusion detection system combining machine learning and principal component analysis (PCA) to increase network security. A network traffic classification system was built and tested on the NSL-KDD dataset and used PCA for dimensionality reduction. The results were cross-validated to reduce overfitting and ensure generalizability of the model. Low-variance precision refers to the consistency of the cross-validation fold. The combination of PCA and machine learning models exceeds previous studies with an F1 score for the random forest model of over 99%. The study improves intrusion detection and network protection against cyber-attacks.**

Keywords: Intrusion Detection System, Principal Components Analysis, NSL-KDD, Machine Learning, Cross Validation.

I. INTRODUCTION

Intrusion Detection Systems (IDS) play a crucial role in contemporary network security frameworks, serving as safeguards against a wide range of potential cyberattacks [1]. These sophisticated technologies continuously monitor the intricacies of networks and systems, serving as the first line of defense. Their primary responsibility is to promptly detect and alert administrators about any security breaches, enabling swift measures to mitigate the impact of cyberattacks. While traditional intrusion detection systems remain crucial to the defensive system, their performance is significantly constrained by the ever-evolving and dynamic nature of cyber threats.

Previous methods such as data encryption, firewalls, and user authentication are insufficient to prevent Internet attacks, so cyber solutions that work to avoid anomalies and attacks [2] came to protect data from attack. Intrusion detection systems monitor network traffic and prevent intrusion. Network intrusions threaten the integrity, confidentiality, and availability of resources [3]. Many network and host-based intrusion detection systems use machine learning to classify and detect cyberattacks.

Principal Component Analysis (PCA) is becoming increasingly recognized as a powerful method in the realm of network security for comprehending patterns in network traffic data. The main purpose of PCA is to streamline analysis by decreasing the number of dimensions in the data, hence facilitating the identification of patterns and anomalies that could indicate intrusions. PCA is utilized in network security to identify anomalies and improve the security stance [4]. Moreover, PCA aids in diminishing noise in intrusion detection data [5], minimizing storage requirements, and accelerating data processing by capturing the majority of the variability in the data using a smaller collection of primary components.

In light of the rising complexity of cyber-attacks and the necessity to tackle issues such as anomaly identification and immediate remediation, it is imperative to expedite the development of a proficient and resilient IDS. Most of the literature tends to prioritize the examination of machine learning and PCA as distinct topics. This study aims to address the existing research void by introducing a novel framework that effectively integrates machine learning and PCA for intrusion detection systems.

II. RELATED WORK

Several recent studies have employed AI techniques, particularly supervised machine learning, to improve the security of smart grids. In their study, the authors of [6] performed a comparison analysis to assess the effectiveness of three supervised techniques - bagging, boosting, and stacking - in detecting cyber-attacks on smart grids. The findings demonstrated that the stacking classifier surpassed other strategies in terms of performance. Authors of [7] have utilized a range of boosting ensembles and standard supervised models to detect breaches in smart grids. Compared to the standard models, the Boosting ensemble classifiers exhibited significantly superior performance. In the same manner, the authors of [8] assessed the efficacy of four established supervised machine learning models in detecting intrusions in smart grids.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 1-7, February-2024

https://doi.org/10.47001/IRJIET/2024.802001

The study conducted in [9] assessed the efficacy of various classification algorithms, and the findings demonstrated the Decision Tree classifier's superiority in identifying intrusions.

Multiple studies have been carried out to employ supervised deep learning methods for the purpose of identifying intrusions in smart grids. The authors in [10] revealed that convolutional neural networks (CNNs) and long short-term memory (LSTMs) are components of the detecting mechanism. In [11], a more advanced supervised convolutional neural network was introduced to detect anomalies in network behavior patterns. A different methodology, as outlined in reference [12], introduced a hybrid framework for detecting unauthorized access in intelligent power grids. The utilization of a Kalman filter in tandem with a recurrent neural network (RNN) was employed in this model. This hybrid model functions at two levels, wherein it makes predictions and fits both linear and nonlinear data. It achieves this by utilizing a fully linked module to merge the outcomes.

Many studies have tested unsupervised cyberattack detection methods. A stack autoencoder identified fake data injection attempts in [13]. K-means data clustering created an external meta-model for smart home-energy center data transfer [14]. In [15], an unsupervised Isolation Forest model was used to identify smart grid hazards. PCA and Isolation Forest were trained, tested, and verified to extract features from unlabeled data. Generative Adversarial Network (GAN)-based anomaly-based intrusion detection was presented [16]. The detection method uses network traffic, TCP, and operational data. These levels spot attacks. A restricted Boltzmann machine identified cyber threats in large smart grid networks [17].

Because it considers informal subsystem interactions, feature extraction and symbolic dynamic filtering minimize computation load. Hierarchical temporal memory was proposed for real-time anomaly detection [18]. Another study [19] identified unsupervised smart grid security hazards using autoencoder and random forest. Malignant vulnerabilities, benign processes, and normal events were well classified by the model. There were some gaps and weaknesses in most previous research, such as detecting zero-day attacks, the complexity of hybrid methods, and others, which must be overcome and building an effective protection system in all circumstances.

III. PROPOSED INTRUSION DETECTION SYSTEM

Intrusion detection systems (IDS) must be developed and implemented to protect networks from threats and attacks in the ever-changing cybersecurity sector. The goal is to create a reliable system that can classify network behavior as normal or symptomatic of various attacks. Figure 1 depicts the NSL-KDD dataset-based IDS development and evaluation technique. Structured data preparation using principal components analysis to minimize dimensionality, train machine learning models, and validate system performance.

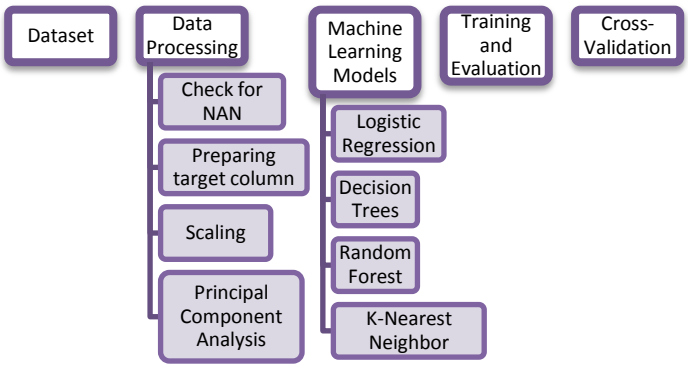

Figure 1: Proposed system steps

The Python environment will be implemented to achieve the objectives of data analysis and machine learning. Initially, the main libraries were imported. The libraries comprised NumPy, Pandas, Matplotlib, Seaborn, TensorFlow, and scikit-learn. Data science and machine learning can leverage the extensive range of capabilities offered by these libraries. These tools have a wide range of applications. In addition, machine learning models will be imported to ensure a holistic approach to building and assessing prediction models.

3.1 Dataset

NSL-KDD is an improved version of the KDD cup99 dataset that seeks to resolve certain problems observed in the previous iteration. This dataset is a significant reference for academics to assess and compare different approaches of IDS, create IDS systems (either host-based or network-based), and conduct experiments in the field of cybersecurity. The dataset has 125,972 records and 43 features (columns) of various categories, as depicted in TABLE 1. The columns in this dataset contain diverse network traffic features, which is typical for datasets used in intrusion detection or network security activities.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 1-7, February-2024

https://doi.org/10.47001/IRJIET/2024.802001

Table 1: List of NSL-KDD attributes

Total Attributes		
Duration	su_attempted	same_srv_rate
protocol_type	num_root	diff_srv_rate
service	num_file_creation	srv_diff_host_rate
flag	num_shells	dst_host_count
src_byte	num_access_file	dst_host_srv_count
dst_byte	num_outbound cmds	dst_host_same_srv_rate
land	is_host_login	dst_host_diff_srv_rate
wrong_fragment	is_gust_login	dst_host_same_src_port_rate
urgent	count	dst_host_srv_diff_host_rate
hot	srv_count	dst_host_serror_rate
num_failed_login	serror_rate	dst_host_srv_serro_rate
logged in	srv_serror_rate	dst_host_rerror_rate
num_compromised	rerror_rate	dst_host_srv_rerror_rate
root_shell	srv_rerror_rate	class

3.2 Data Processing

Several operations were performed to process the data and make it suitable for classification algorithms, such as:

- Check for NAN: The dataset has no missing values (NaN) according to data.isnull().sum(). Data quality is critical since missing values can dramatically impact machine learning model performance and accuracy.
- Preparing target column: The Target field categorizes network activities as either assaults or normal traffic. The number of unique values was determined in the "Target" column, and the frequencies of these distinct values were calculated. All forms of attacks were consolidated into a single category, referred to as "attack," and the problem was addressed as a binary classification task, distinguishing between normal and attack instances. Figure 2 displays the distribution of the "target" variable following its conversion into a binary classification issue.
- Scaling: RobustScaler from scikit-learn was used to measure numerical properties. This activity is beneficial when working with data sets with unusual data points. The 'Target' field has binary values: 0 for 'Normal' results and 1 for all others. Three categorical columns—"protocol_type," "service," and "flag," were encrypted using one-hot encoding to make them machine learning-friendly. Resulting data frame size were 125972 and 124.
- Principal Component Analysis: PCA was utilized to decrease the dimensionality of the feature matrix. The specified number of components was determined using the n_components argument. Furthermore, the target labels were transformed into integers. The original and reduced feature matrices and target labels were split into training and testing sets. Data was split 20% for testing and 80% for training.

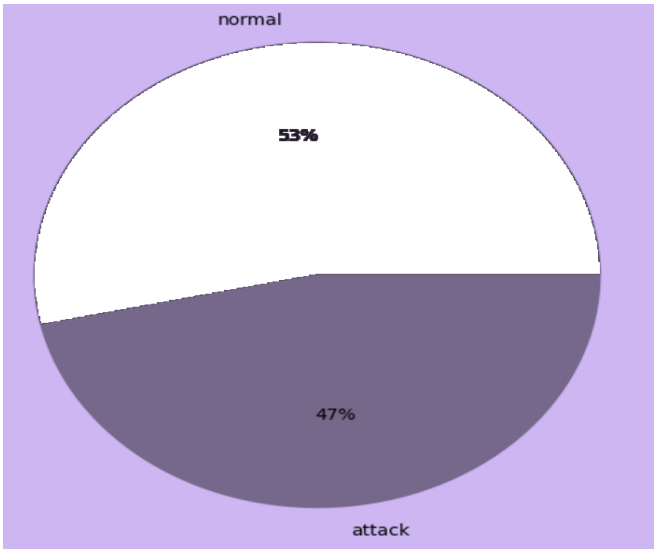

Figure 2: Distribution of the target column

3.3 Machine Learning Models

When it comes to this classification challenge, various machine learning models have been configured in order to achieve the best possible outcomes, which are as follows:

- Logistic Regression: This popular classification method is ideal for binary and multi-class categorization. The method estimates class membership using a logistic function. It then calculates event log probability using the logistic function. For linearly separated data, the method is simple, effective, and clear.
- Decision Trees: Decision trees are hierarchical computational models with internal nodes representing characteristics and terminal nodes representing conclusions. An iterative technique divides the data by the most important attribute at each iteration, improving class differentiation or target variable prediction. The method is easy to interpret and can handle numerical and categorical data without preprocessing.
- Random Forest: Ensemble learning trains numerous decision trees and calculates the class pattern (for classification) or average forecast (for regression) from them. The data and features used to train each tree are random. This system excels in precision, robustness in excessive data manipulation, and effectiveness in managing multidimensional datasets.
- K-Nearest Neighbor: The straightforward and intuitive K-NN algorithm is used for classification and regression. Classifying the closest neighbors in feature space involves determining their dominant class. This is done by calculating the distance between the input instance

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 1-7, February-2024

https://doi.org/10.47001/IRJIET/2024.802001

and all dataset examples. The majority or average vote of the nearest neighbors determines class or value. The approach is easy to understand, versatile, and successful in low-dimensional environments.

3.4 Training and Evaluation

The models' performance was evaluated using key evaluation metrics such as:

- Accuracy: The ratio of accurately predicted instances to total instances.
- Precision: This is the ratio of real positive forecasts to overall positive predictions.
- Recall: It is the ratio of genuine positive predictions to actual positives.
- F1-Score: The average harmonic of precision and recall offers a balanced model efficiency measure.
- ROC curve: It shows the model's performance at different categorization levels. At varied thresholds, it graphs recall versus false positive rate. The area under the ROC curve (AUC-ROC) determines performance, and a model with a score near to 1 can discriminate positive and negative cases.

A logistic regression model is created and saved, ensuring the parameter random_state=1 is a seed for generating random integers. The model is trained and the trained model is then used to generate predictions on the test data. The ROC curve resulting from the test is shown in Figure 3.

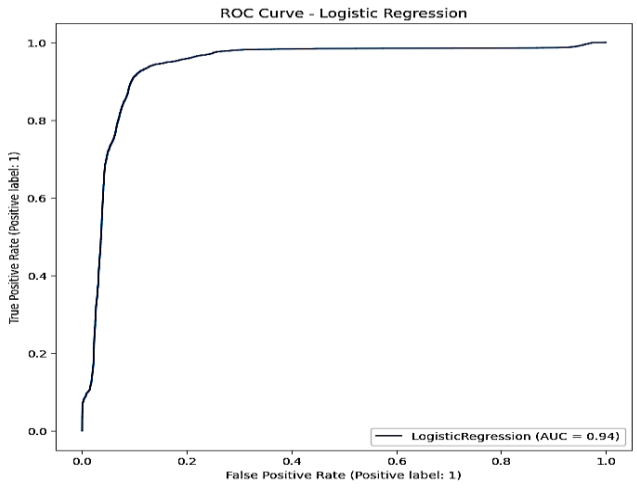

Figure 3: ROC curve of logistic regression

A decision tree model is constructed with the option max_depth=3, which specifies the maximum number of levels in the tree. This parameter is set to prevent overfitting by restricting the depth of the tree. When utilizing test data to make predictions, the ROC curve yielded the outcome depicted in Figure 4.

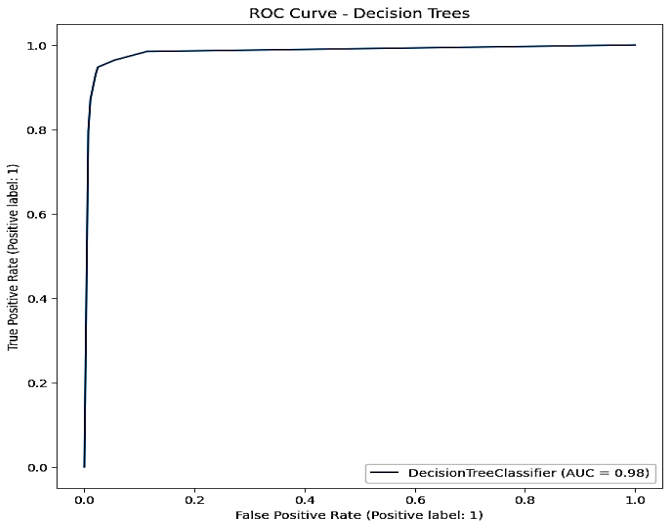

Figure 4: ROC curve of decision trees

Random Forest builds many decision trees and blends their forecasts to improve accuracy and robustness. Create, save, and train a random forest model. The trained model then predicts test data. Figure 5 shows the ROC curve.

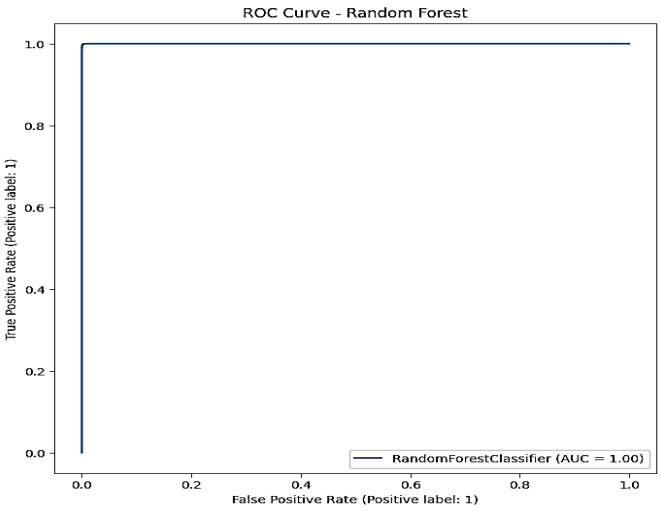

Figure 5: ROC curve of random forest

The K-Nearest Neighbors algorithm classifies instances quickly and easily. It classifies an instance by evaluating the majority class of its feature space closest neighbors. A K-Nearest Neighbors model is created, stored, and trained with n_neighbors=20, which specifies the number of neighboring data points to consider during classification. Figure 6 shows the ROC curve from trained model predictions on test data.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 1-7, February-2024

https://doi.org/10.47001/IRJIET/2024.802001

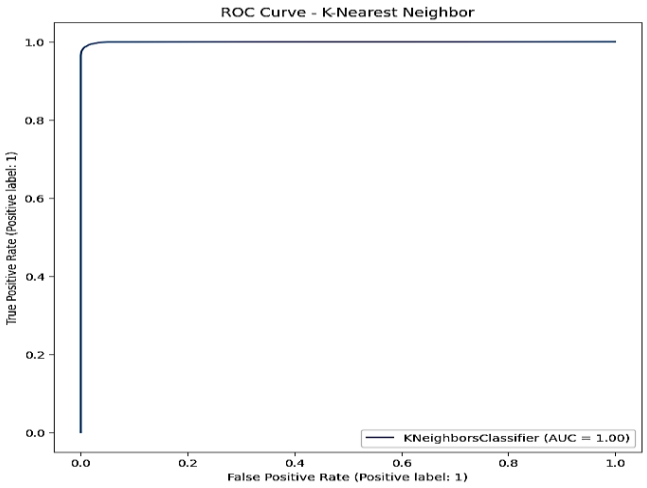

Figure 6: ROC curve of k-nearest neighbor

Table 2 presents the performance results of each model in handling the distributions within the dataset. The random forest model is renowned for its exceptional accuracy and near-perfect results.

Table 2: Evaluation results

	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.903274	0.869093	0.934372	0.900551
Decision Trees	0.962373	0.971602	0.947413	0.959355
Random Forest	0.998333	0.999321	0.997121	0.998220
K-Nearest Neighbor	0.989403	0.990648	0.986705	0.988673

Each model performs very well when it comes to dealing with an imbalanced data set, a situation where it is necessary to maintain a balance between precision and recall. After the Random Forest model, which achieved the highest F1 score of 0.998220, the K-Nearest Neighbor model came in second place.

3.5 Cross-Validation

It is a method that entails iteratively training and testing machine learning models on distinct subsets of the given data to assess their performance. Cross-validation provides a more reliable evaluation of model performance by averaging outcomes across multiple folds, hence assisting in the detection of overfitting to the training data. Validation was conducted on each model using both the training and test sets. The results were saved and the mean accuracy and standard deviation were reported, as depicted in Figure 7. Validation enables a thorough examination of the accuracy of all models on both training and test sets, allowing for comparison and assessment of the models' ability to generalize.

```
Logistic Regression Cross-Validation Accuracy (Training Set): 89.80% (± 0.29%)
Logistic Regression Cross-Validation Accuracy (testing Set): 89.92% (± 0.55%)
Decision Trees Cross-Validation Accuracy (Training Set): 96.45% (± 0.17%)
Decision Trees Cross-Validation Accuracy (testing Set): 96.19% (± 0.29%)
Random Forest Cross-Validation Accuracy (Training Set): 99.83% (± 0.02%)
Random Forest Cross-Validation Accuracy (testing Set): 99.65% (± 0.10%)
K-Nearest Neighbor Cross-Validation Accuracy (Training Set): 98.84% (± 0.04%)
K-Nearest Neighbor Cross-Validation Accuracy (testing Set): 98.13% (± 0.19%)
```

Figure 7: Validation results

IV. RESULTS DISCUSSION

Our model will be evaluated by comparing its performance with that of previous studies that used NSL-KDD which is widely used as shown in TABLE3. The autoencoder model developed by Wen Xu et al. [19] Network anomaly detection is a sophisticated 5-layer architecture. The model uses innovative technology to prepare data and efficiently handle imbalances. Tongtong Su et al. presented the BAT model [20], which integrates attention and BLSTM. The attention mechanism examines packet vectors generated by the BLSTM model to classify network traffic. Data samples were processed using multiple convolutional layers, and network traffic was classified using softmax. Farahat et al. examined the latest concepts in security intrusion detection in [21], with special emphasis on emerging patterns and innovative contributions. The dataset and its various components were evaluated by WEKA, and the Random Forest technique yielded the best classification results.

Table 3: Comparing the findings with similar studies

Ref.	Method	Results
[19]	5-layer autoencoder (AE)-based model (Unsupervised feed-forward neural network)	Accuracy 90.61% F1-score 92.26%
[20]	Deep Learning (BAT model combines BLSTM and attention mechanism)	Accuracy 84.25%
[21]	SVM, k-NN, Naïve Bayes, Logistic Regression, Random Forest, Decision Trees	F1-score 98.6%
Proposed Model	PCA + ML Models	Accuracy 99% F1-score 99%

Combining principal components analysis with traditional machine learning models showed excellent results and metrics including precision, recall, and F1 score yielded

ISSN (online): 2581-3048
Volume 8, Issue 2, pp 1-7, February-2024
https://doi.org/10.47001/IRJIET/2024.802001

positive results within the dataset. The results were validated and all models showed strong predictive capabilities, with Random Forest being the best. The use of principal components analysis in typical machine learning models has been effective in intrusion detection systems.

V. CONCLUSION

Conventional intrusion detection methods encounter difficulties in handling the extensive and constantly evolving characteristics of contemporary network threats. Machine learning provides flexibility and autonomous learning capabilities, whereas PCA gives a method to decrease features and enhance model efficiency. The objective of this project is to integrate these methodologies to create an advanced IDS framework capable of accurately detecting and categorizing intrusions in intricate network environments. A network traffic detection and classification algorithm were developed for the NSL-KDD dataset. The data was prepared and processed, and then PCA was applied to minimize its dimensionality. Subsequently, four machine learning models (logistic regression, decision trees, K-nearest neighbor, random forest) were trained specifically for binary classification. When tested, the technique produced remarkable outcomes across a range of performance metrics. The models' generalizability and absence of overfitting were checked by cross-validation. Performance was consistent throughout several cross-validation folds, as evidenced by the small variance in accuracy. With an F1 score of over 99% and an astounding accuracy rate, the suggested method proved to be more effective than prior research in detecting and classifying intrusions. Contributing substantially to ongoing efforts to enhance network security against dynamic cyber threats, the study offered a better architecture for detecting breaches in network security. In future work, we propose to increase the representation of network data by incorporating domain expertise and extracting critical features that reflect precise invasion patterns using advanced feature engineering methods. Real-time IDS models can also be developed to adapt to changing network conditions and adapt to new threats without retraining.

REFERENCES

[1] D. Kapil, N. Mehra, A. Gupta, S. Maurya, and A. Sharma, "Network security: threat model, attacks, and IDS using machine learning," *in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 203-208, 2021.

[2] A. D. Jadhav, "Two Phase-Intrusion Detection System (TP-IDS) model using Machine Learning Techniques," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 9, pp. 417-425, 2021.

[3] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, vol. 28, Suppl 1, pp. 1051-1058, 2017.

[4] B. Wen and G. Chen, "Principal component analysis of network security data based on projection pursuit," *in International Conference on Network Computing and Information Security, Berlin, Heidelberg: Springer Berlin Heidelberg*,pp. 380-387, 2012.

[5] K. K. Vasan and B. Surendiran, "Dimensionality reduction using principal component analysis for network intrusion detection," *Perspectives in Science*, vol. 8, pp. 510-512, 2016.

[6] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble Learning Methods for Anomaly Intrusion Detection System in Smart Grid," in Proc. *2021 IEEE International Conference on Electro Information Technology (EIT), Mt. Pleasant, MI, USA*, pp. 129-135, 2021.

[7] T. T. Khoei, S. Ismail, and N. Kaabouch, "Boosting-based Models with Tree-structured Parzen Estimator Optimization to Detect Intrusion Attacks on Smart Grid," in Proc. *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0165-0170, 2021.

[8] [8] Z. E. Mrabet, H. E. Ghazi, and N. Kaabouch, "A performance comparison of data mining algorithms-based intrusion detection system for smart grid," *in Conference on Electro Information Technology (EIT), IEEE, Piscataway, NJ, USA*,pp. 298-303, 2019.

[9] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *Internet of Things Journal*, vol. 6, pp. 9042-9053, 2019.

[10] R. Yao, N. Wang, Z. Liu, P. Chen, and X. Sheng, "Intrusion Detection System in the Advanced Metering Infrastructure: A Cross-Layer Feature-Fusion CNN-LSTM-Based Approach," *Sensors*, vol. 21, 626, 2021.

[11] H. Yang and F. Wang, "Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network," IEEE Access, vol. 7, pp. 64366-64374, 2019.

[12] Y. Wang, Z. Zhang, J. Ma, and Q. Jin, "KFRNN: An Effective False Data Injection Attack Detection in Smart Grid Based on Kalman Filter and Recurrent Neural Network," *IEEE Internet of Things Journal*, vol. 9, pp. 6893-6904, 2022.

[13] S. Majidi, S. Hadayeghparast, and H. Karimipour, "FDI attack detection using extra trees algorithm and deep

ISSN (online): 2581-3048
Volume 8, Issue 2, pp 1-7, February-2024
https://doi.org/10.47001/IRJIET/2024.802001

learning algorithm-autoencoder in smart grid," *International Journal of Critical Infrastructure Protection*, vol. 37, 100508, 2022.

[14] S. Ahmed, Y. Lee, S. Hyun, and I. Koo, "Unsupervised Machine Learning-Based Detection of Covert Data Integrity Assault in Smart Grid Networks Utilizing Isolation Forest," *IEEE Transactions on Information Security*, vol. 14, pp. 2765-2777, 2019.

[15] D. M. Menon and N. Radhika, "Anomaly detection in smart grid traffic data for home area network," in Proc. *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Nagercoil, India, pp. 1-4, 2016.

[16] P. R. Grammatikis, P. Sarigiannidis, G. Efstathopoulos, and E. Panaousis, "ARIES: A Novel Multivariate Intrusion Detection System for Smart Grid," *Sensors*, vol. 20, 5305, 2020.

[17] H. Karimipour, A. Dehghantanha, R. M. Parizi, K. R. Choo, and H. Leung, "A Deep and Scalable Unsupervised Machine Learning System for Cyber-Attack Detection in Large-Scale Smart Grids," *IEEE Access*, vol. 7, pp. 80778-80788, 2019.

[18] A. Barua, D. Muthirayan, P. P. Khargonekar, and M. A. Al Faruque, "Hierarchical Temporal Memory Based Machine Learning for Real-Time, Unsupervised Anomaly Detection in Smart Grid: WiP Abstract," in Proc. *ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, Sydney, Australia, pp. 188-189, 2020.

[19] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on NSL-KDD dataset," *IEEE Access*, vol. 9, pp. 140136-140146, 2021.

[20] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575-29585, 2020.

[21] S. Farhat, M. Abdelkader, A. Meddeb-Makhlouf, and F. Zarai, "Comparative study of classification algorithms for cloud IDS using NSL-KDD dataset in WEKA," *in 2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 445-450, 2020.

Citation of this Article:

Rula Abdulwahid Mohammed, Youssef A. Bazzi, "Implement an Intrusion Detection System Utilizing Machine Learning and Principal Component Analysis" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 8, Issue 2, pp 1-7, February 2024. Article DOI https://doi.org/10.47001/IRJIET/2024.802001

****

