International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.
IJSRA
International Journal of Science and Research Archive
Research Journal Archive, INDIA

(Review Article)

Mitigating insider’s threats using support vector machine and k-nearest Neighbour

Maureen I. Akazue *, Nkiru Queen Muka and Abel E. Edje

Department of Computer Science, Delta State University, Abraka, Nigeria.

International Journal of Science and Research Archive, 2024, 12(01), 2626–2635

Publication history: Received on 08 May 2024; revised on 17 June 2024; accepted on 20 June 2024

Article DOI: https://doi.org/10.30574/ijsra.2024.12.1.1110

Abstract

Addressing insider’s threats is a critical challenge in organizational security. This study presents the development and evaluation of a hybrid machine learning model aimed at enhancing insider’s threat detection effectiveness. The escalating risks associated with insider’s threats necessitated advance detection mechanisms to mitigate potential breaches. Leveraging the strengths of multiple individual models, including Support Vector Machine (SVM) and K-nearest Neighbour (KNN), the hybrid model addressed this challenge by improving detection accuracy while minimizing false positives. Through rigorous evaluation, the hybrid model demonstrates remarkable performance, achieving an accuracy of 99%, with precision, recall, and F1 score of 99%, 98%, and 97% respectively. By providing a robust solution to insider’s threat detection, the hybrid model offers organizations a promising approach to fortify security measures and safeguard against potential breaches.

Keywords: Support Vector Machine (SVM); K-Nearest Neighbour (KNN); Hybrid Model, Machine Learning; Insider’s Threat and Jupyter Notebook

1. Introduction

Attacks on cybersecurity are occurring more often which presents worrisome dangers in a variety of industries. These days, it's difficult to identify these attacks because they take advantage of emerging technologies. Protecting people and businesses from these types of criminal activity is the main goal of cyber security, which also tries to lessen the possibility of damage being done to computer networks, resources, applications, and data. Data, host, network, and application security are only a few of the layers at which protective measures are implemented [1], [2], [3]. Remarkably, assessments conducted by the industry indicate that an astounding 79% of security threats come from insiders. Finding authorized individuals who may be harming the organization even though they are trustworthy is the most difficult cyber security assignment [4], [5].

One of the biggest problems in cybersecurity is insider attacks. Insiders operate within the system like normal users, which adds complexity to the process of identifying and categorizing them. Nevertheless, there is still a lack of contemporary application of sophisticated machine learning techniques in this field [6], [8]. The term insider’s threat refers to the deliberate or unintentional misuse of authorized access to a company's network, system, or data by workers, contractors, or business partners; this could have a negative impact on the availability, confidentiality, or integrity of the company's information systems [9], [11].

Insider’s threats can have serious repercussions, such as monetary losses, reputational harm, and legal obligations. Using machine learning to identify insider’s risks is one promising strategy. Large data sets can be analyzed by machine learning algorithms, which can then spot trends and abnormalities that might point to threats [12], [13], [14]. But machine learning algorithms might not be enough on their own to identify every kind of insider’s threat. As a result, a hybrid model that incorporates additional methods into machine learning may be more successful in identifying

[*] Corresponding author: Maureen I. Akazue

insider's risks. This hybrid model will incorporate anomaly detection mechanism, rule-based systems, and analytics on user behavior with machine learning.

2. Review of Related Literature

Insider's threats have become one of the biggest obstacles that organizations face when trying to protect their sensitive data and information in recent years. In the study carried out by [15], it provides a thorough analysis of various machine learning algorithms and how well they detect insider risks. For increased detection accuracy, the study highlights how crucial it is to choose the right characteristics and classifiers. The knowledge gained from this study will be helpful in creating hybrid machine learning models that will improve the ability to detect threats [16]. The researchers conducted experiments and created a feature vector by removing risk indicators from the CERT insider's threat dataset. In the work of [17], [18], they provided a helpful advice on how to stop, identify, and handle insider's threats. The guide provides practical tactics for reducing insider-related crimes, like as theft, sabotage, and fraud, based on actual case studies. In the work of [19], they used iForest as an unsupervised anomaly detection technique. Their main goal was to identify statistically anomalous behavior by examining several features taken from social data, like online activity and email communication patterns. In the field of security, formal methods have become more popular. As proposed by [20], an assessment of different formal methods and instruments for insider's threat identification as well as modeling and confirming security attributes. Researchers and practitioners interested in using formal approaches to improve insider's threat prevention will benefit greatly from the advice provided by this survey. In the work of [21] which involves a bio-inspired auto-resilient policy regulation framework. Their strategy combines biologically inspired mechanisms with formal ways to improve system resilience against insider's attacks. As opined by [22], they concentrated on using iForest for an online insider's threat detection. It created a generic algorithm to determine which HTTP features work best for differentiating abnormal insider's behaviors from normal ones

Several machine learning (ML) techniques, such as supervised, unsupervised, and semi-supervised learning approaches, have been used to identify insider's risks [23]. Based on labeled training data, supervised learning algorithms like Random Forests and Support Vector Machines (SVM) have been utilized to categorize user behavior as harmful or benign [24]. Unusual patterns in data have been found using unsupervised learning approaches like anomaly detection and clustering, even in the absence of prior knowledge of malicious activity [2], [3], [5], [25]. In situations with little labeled data, semi-supervised learning techniques that integrate labeled and unlabeled data have been investigated to enhance detection performance [26]. As proposed by [27], anomaly detection of insider's actions by multi-class classification. The researchers explore methods to distinguish normal behaviours from anomalous ones, enabling more accurate identification of potential insider's threats. Their findings highlight the effectiveness of multi-class classification techniques in enhancing insider's threat detection capabilities. In the work of [28], they looked into the application of Light Gradient Boosting Machine (LightGBM) for insider threat detection in a different study. Because LightGBM is adept at managing intricate and multidimensional data, it is a good fit for examining user behavior patterns linked to insider's threats. An approach to insider's threat identification based on user behavior analysis is presented by [29]. They employed machine learning techniques to examine user activities and identify trends that deviate from typical behaviour. By observing user's behavior and system interactions, this behavioral analysis approach offers insights into spotting questionable insider's activity [30-31]. It was introduced in the survey carried out [32]. It was observed that the accuracy of insider's threat prediction is improved by LSTM's capacity to identify patterns and temporal dependencies in log data. The work demonstrates how LSTM-based models can handle sequential data for insider's threat detection in an efficient manner.

In the study carried out by [33], they introduced a unique method for detecting insider's threats by using a Convolutional Neural Network (CNN) that was trained from scratch to distinguish between colored images that depict malicious and benign activity. A constraint learning method was used to identify insider's threats. By building an optimized constraint network that simulates typical behavior, this system detects hazards by calculating the cost that exceeds a predetermined threshold [34]. Unusual behavior suggestive of insider's threats can be identified by machine learning algorithms, especially using anomaly detection techniques [35]. These algorithms can identify unusual patterns in user behavior, access privileges, and efforts at data exfiltration by utilizing neural networks, clustering, and classification.

In order to improve the precision and resilience of insider's threat detection systems, researchers have emphasized the significance of combining a variety of data sources, including as network logs, user activity logs, and endpoint data [36]. Machine learning algorithms are able to detect insider's threats by capturing their complex character through the use of many data sources. Insider's threat detection has become a popular application for explainable machine learning models [37]. These models offer clear insights into the decision-making process, making it easier for security analysts to comprehend the reasoning behind the detection of possible insider threats and to develop more strong response plans. The efficacy of anomaly detection algorithms was exhibited in the work of [38], who were able to identify anomalous patterns that could be signs of insider's threats with high accuracy. For insider's threat detection, [39] stressed the significance of combining various data sources. They discovered that when machine learning models were used to leverage a variety of data sources, such as network logs, user activity logs, and endpoint data, the accuracy of the models improved to over 95%. According to [40], they explained different machine learning algorithms are highly accurate in identifying insider's threats.

As proposed by [41], they assessed that machine learning models has better accuracy rate in detecting insider's threats when compared with other existing models. The study showed that using support vector machines and deep learning models may achieve precision and recall rates of more than 85%, proving the effectiveness of these methods in precisely detecting insider's threats. Also, [42] reemphasized the high degree of accuracy attained in threat identification when machine learning and natural language processing techniques are used.

3. Methodology Adopted

By employing a hybrid paradigm, the suggested system developed an advanced insider's threat detection system. The datasets were properly prepared before any machine learning method can be used. To ensure that the raw data is appropriate for the learning algorithm, data was cleaned and preprocessed. Feature extraction was done to uncover latent patterns that acted as independent variables for the algorithm to learn once the dataset has been cleaned and preprocessed. Afterwards, sophisticated machine learning techniques were utilized to create a model that precisely categorize certain data points as either harmful or routine actions. Evaluation criteria including accuracy, recall, precision, and F1-Score was used to gauge the model's accuracy.

4. The Architecture System

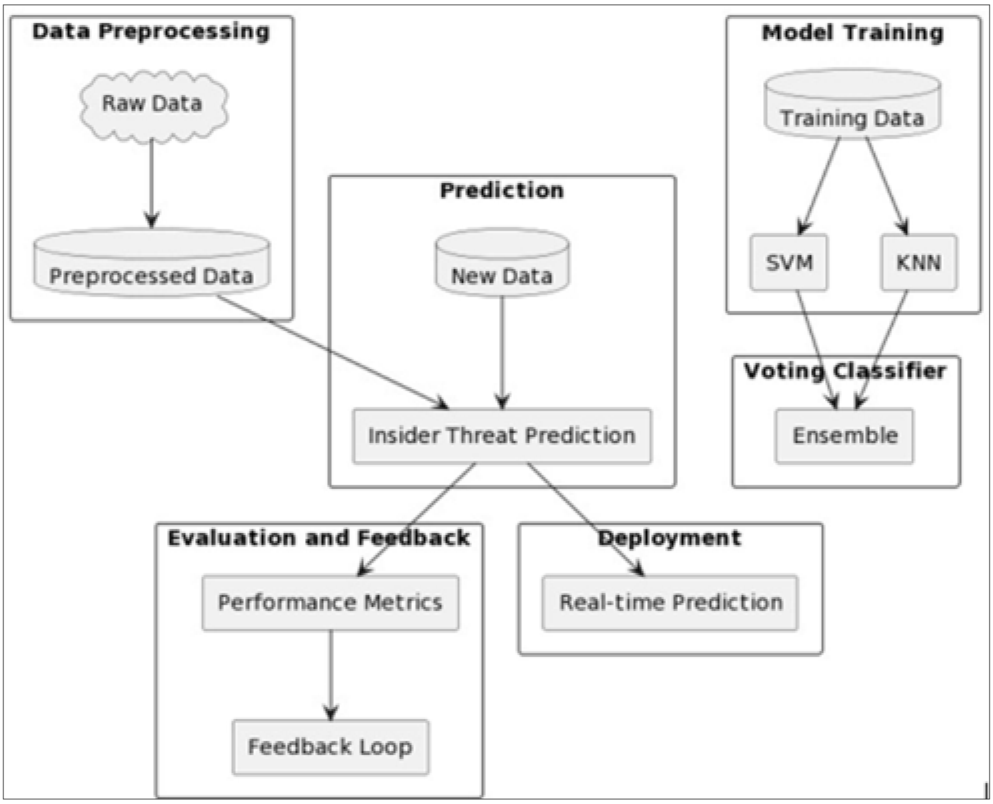


Figure 1 Architecture of Proposed System

The system's overall architectural design for insider's threat detection is referred to as the High-Level Model of the Proposed System. The main elements of the system are delineated, along with their interrelationships. This affords an overview of how different modules, including data preprocessing, model training, ensemble modeling, prediction, evaluation, and deployment, collaborate to accomplish the goal of identifying insider's threats. The development and integration of the various components to produce a coherent and successful insider's threat detection solution are guided by this high-level model, which acts as a blueprint for the system's functionality and structure. This system has the following distinct advantages; Enhanced Accuracy through Model Fusion, Complementary Model Capabilities, Effective Handling of Heterogeneous Data, Improved Interpretability, Optimized Resource Utilization and Robust against over fitting. Figure 1 shows the interaction between the components of the modeled system

5. The Analysis of the System

Generally, the system is built combining the strength of the following models, Support Vector Machines (SVM), K-Nearest Neighbors (KNN) and Hybrid Model. Support Vector Machines (SVM) is a robust supervised machine learning algorithm designed for classification and regression tasks. The machine learning algorithm K-Nearest Neighbors (KNN) is a flexible tool that predicts things by comparing data points. KNN learns the entire dataset during the training phase while the is an amalgamation or integration of many methods or models that capitalize on their unique advantages while mitigating their drawbacks. The objective is to build a more powerful and efficient model that performs better than the sum of its components. The system has the following components;

- **Data Preprocessing:** Relevant features are extracted from the raw data to get it ready for model training.
- **Model Training:** Using the preprocessed data, SVM and KNN models are trained to find patterns and relationships in the data.
- **Voting Classifier:** To arrive at a final prediction, the trained SVM and KNN models' predictions are combined using a Voting Classifier.
- **Prediction:** In order to detect insider's threats, new data is subjected to predictions made by the combined Voting Classifier model.
- **Evaluation and Feedback:** Performance indicators are utilized to assess the predictions, and the evaluation's feedback can be applied to either the ensemble method or the individual models in the Voting Classifier.
- **Deployment:** The hybrid model can be used to detect insider's threats in real-time by making predictions on fresh data after it has been trained and assessed.

6. Dataset and Data Preprocessing

The CERT Coordination Center, a part of Carnegie Mellon University's Software Engineering Institute, processed and provided the network security data that makes up the CERT 4.2 dataset. The aforementioned source provided the dataset that were used in this investigation. The raw input files from CERT 4.2 were imported and processed during the data processing step. Owing to the high memory needs and quantity of the dataset, example files were created for every category of CSV files. The date values were initially collected in their original format, but they were broken down and encode them into two separate features: day and time. Machine learning algorithms require numerical inputs, thus, the date and time id converted into numerical representations.

Table 1 Encoded features at pre-processing phase

Feature	Possible value
Day	0,1,2,3,4,5.
Time	1,2,3,4……..24
User	String
PC	String
Activity$^{(1)}$	1,2,3,4,5,6,7

7. SMOTE (Synthetic Minority Oversampling Technique)

In addressing the challenge of class imbalance, particularly when dealing with datasets where the occurrences of malicious activities are notably lower than normal ones, the application of SMOTE (Synthetic Minority Over-sampling Technique) proves instrumental. By synthetically generating instances for the minority class, SMOTE effectively rebalances the dataset, ensuring a more equitable representation of both classes. This approach facilitates a more comprehensive learning experience for machine learning models, enabling them to discern patterns and make predictions with greater accuracy across both normal and malicious activities. Thus, the iapplication of SMOTE serves as a crucial step in enhancing the robustness and reliability of the models when faced with class imbalance issues.

7.1. Algorithm 1

- Step1. Load Data from CSV file
- Step 2. Prepare Data
  - Separate features (X) and target variable (y)
  - Split data into training and testing sets
- Step 3: Train Individual Models
  - Decision Tree: Create and train model
  - Random Forest: Create and train model
  - k-Nearest Neighbours: Create and train model
  - Support Vector Machine (SVM): Create and train model
- Step 4. Create and Train a Voting Classifier
  - Combine models into a Voting Classifier with hard voting
  - Train the Voting Classifier
- Step 5. Evaluate Models
  - For each model Evaluate accuracy, recall, precision, and F1 score
  - Plot confusion matrix.
- Step 6. Interpret and Present Results
  - Provide a straightforward indication of model accuracy and performance
  - Optionally, include any additional interpretation or visualization

The activity diagram in figure 2 shows the processes that occurs in the developed system

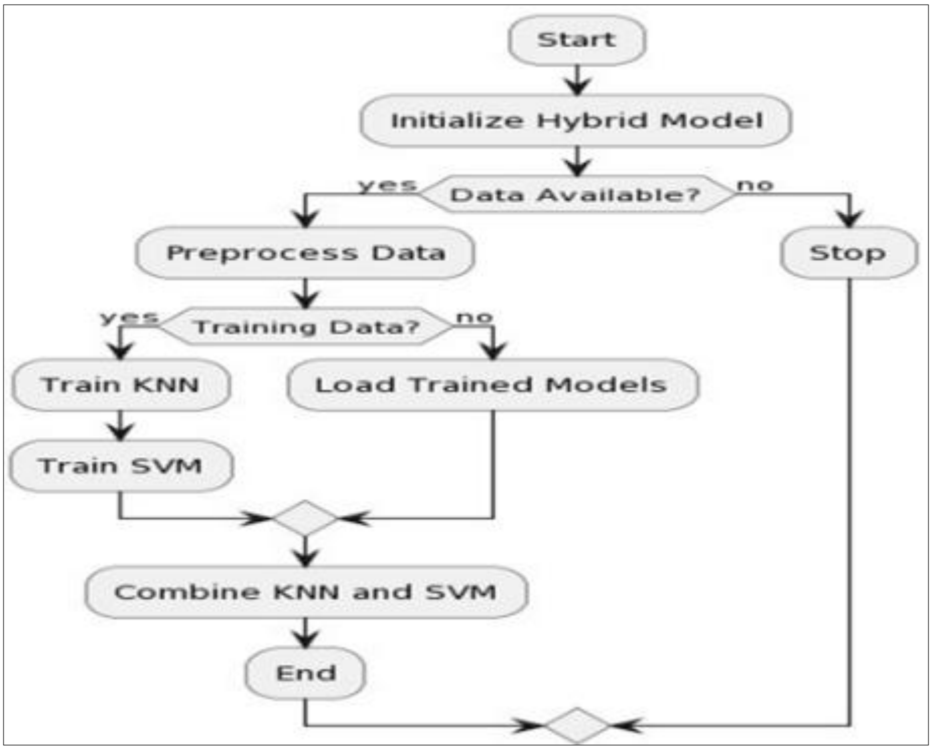


Figure 2 Activity Diagram for the hybrid process

8. Result and Discussion

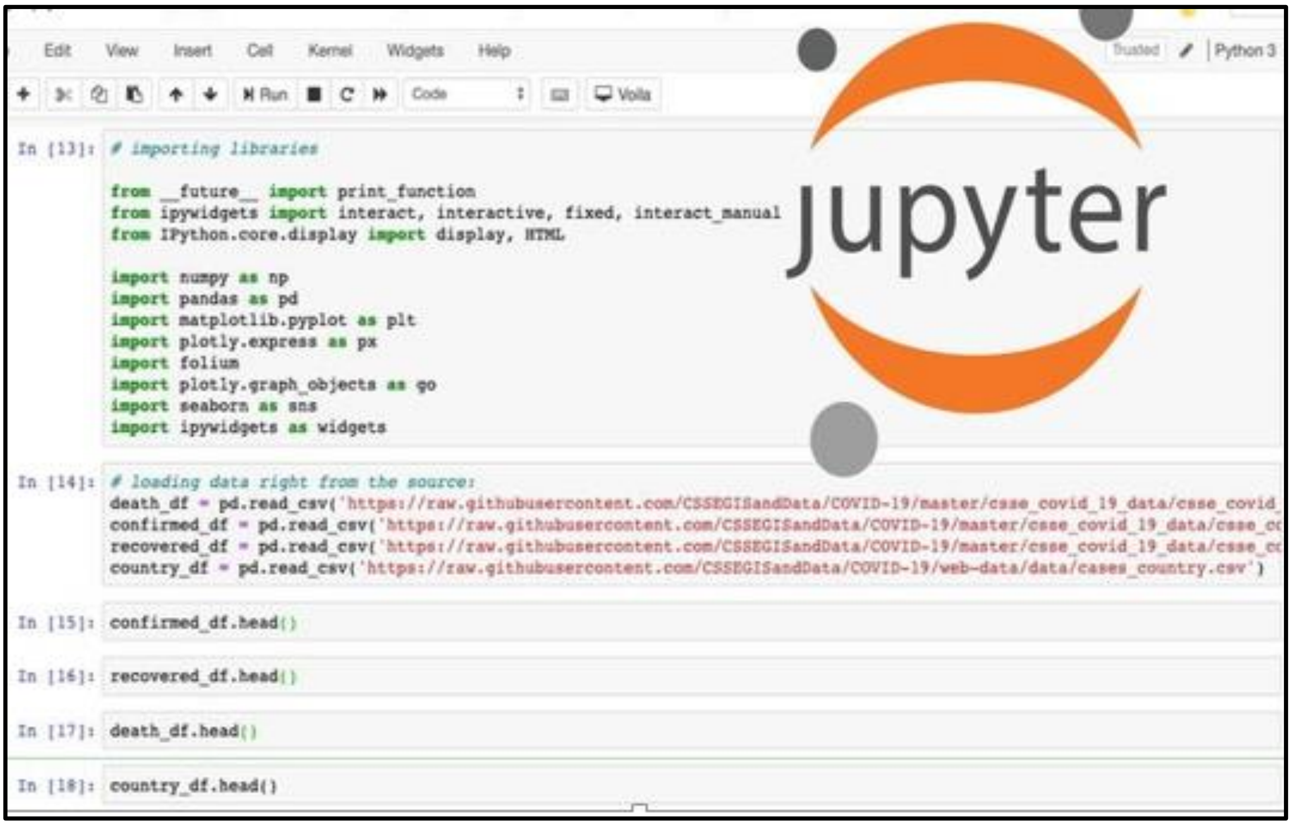
8.1. Developed System Requirements

The hardware and software components that have already been determined and are in use for the successful operation of the implemented system are listed in the suggested System Requirements section. These specifications acted as the cornerstone for the stages of planning, designing, and developing. An explanation of the hardware and software components can be found below.

8.2. Hardware Requirements and Software Requirements

The hardware requirements for this system includes the following; CPU: Intel Core i5 or equivalent, RAM: 8 GB minimum, Storage: 256 GB SSD, GPU (optional): NVIDIA GeForce GTX 1050 or equivalent for accelerated computation while the software requirement for the system; Operating System: Windows 10, macOS, or Linux, Python 3.8, Scikit-learn library for machine learning, Pandas library for data manipulation, Numpy library for numerical computations and Jupyter Notebook.

Jupyter Notebook served as the foundational platform for developing the hybrid model. The Python programming language was our first choice for creating this model because of its versatility and superior ability to handle scientific, statistical, and mathematical procedures. The Jupyter Notebook IDE, which enables the production of interactive and shareable notebooks, was used to build the Python source code. Code, annotations, graphics, and multimedia can all be included in these notebooks. Variety of machine learning and data science packages used includes Seaborn, NumPy, Pandas, Scikit-Learn, and Matplotlib, to construct the model.



```
In [13]: # importing libraries

         from __future__ import print_function
         from ipywidgets import interact, interactive, fixed, interact_manual
         from IPython.core.display import display, HTML

         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import plotly.express as px
         import folium
         import plotly.graph_objects as go
         import seaborn as sns
         import ipywidgets as widgets

In [14]: # loading data right from the source:
         death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid
         confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cc
         recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cc
         country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')

In [15]: confirmed_df.head()

In [16]: recovered_df.head()

In [17]: death_df.head()

In [18]: country_df.head()
```

Figure 3 An overview of Jupyter Notebook

8.3. The System Performance Metrics

The integration of discussions and conclusions were moved through this section advances the field's academic discourse and advances research.

Table 2 Performance Metrics

Model	Accuracy	Precision	Recall	F1 score
Support vector machine	67%	84%	67%	72%
K-nearest Neighbor	89%	93%	89%	90%
Hybrid model	99%	99%	98%	97%

The summary of the Confusion Matric of the three models used in building this system (Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Hybrid Model) are summarized in the performance metrics shown in the bar chart shown below. The Support Vector Machine (SVM) model demonstrates moderate accuracy at 67%, with a precision of 84%, indicating its ability to accurately classify instances it predicts as insider's threats. However, its recall and F1 score, at 67% and 72% respectively, suggest that it may miss a considerable number of actual insider's threats and misclassify some non-threat instances. Conversely, the K-nearest Neighbor (KNN) model achieves a notably higher accuracy of 89%, with strong precision, recall, and F1 score values of 93%, 89%, and 90% respectively, indicating its ability to classify both insider's threats and non-threat instances accurately. Remarkably, the Hybrid model, which likely combines multiple algorithms, outperforms both individual models significantly, exhibiting an outstanding accuracy of 99%, precision of 99%, recall of 98%, and F1 score of 97%. This suggests that the Hybrid model provides the most reliable and consistent predictions, making it the preferred choice for insider's threat detection due to its high accuracy and ability to effectively identify potential threats while minimizing false positives. The Support Vector Machine (SVM) model exhibits the poorest performance among the three models evaluated, with a notably high number of false positives (574). This indicates that SVM misclassifies a significant number of normal instances as insider's threats, resulting in a lower precision score and overall accuracy compared to the other models (See figure 4).

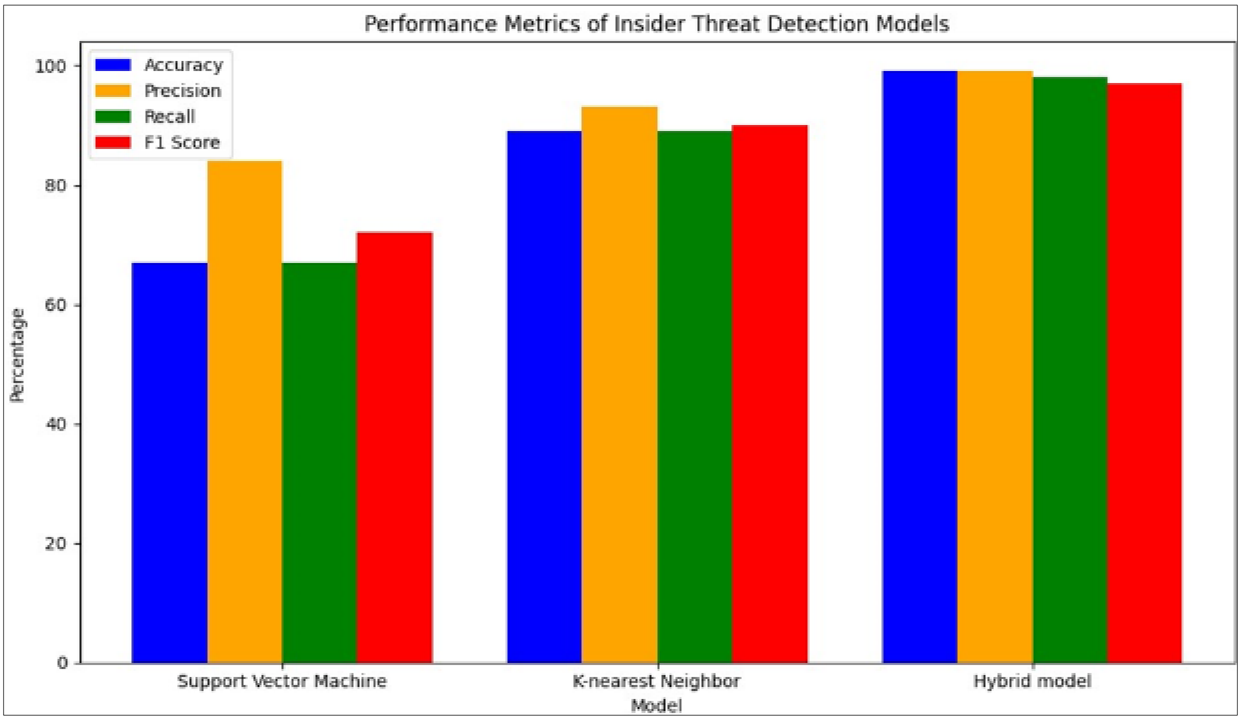


Figure 4 Bar Chart for Performance Metrics

8.4. Result Findings

SMOTE works by generating synthetic samples for the minority class through interpolation between existing instances. This technique effectively balanced the dataset by increasing the representation of the minority class, thus reducing the model's bias towards the majority class and mitigating the tendency to misclassify negative instances as positive.

To streamline feature conversion processes, the hybrid model implemented optimization strategies such as algorithmic enhancements and parallel processing. Additionally, feature engineering techniques were utilized to extract more informative features and reduce dimensionality, thereby expediting the conversion of features into fixed-size matrices. These optimizations resulted in faster processing times, crucial for timely detection of insider's threats and improving overall system efficiency.

Diversifying the dataset beyond the CERT dataset's HTTP log data involved several steps. By integrating multiple data sources, the hybrid model achieved a more comprehensive representation of real-world scenarios, enhancing its adaptability and generalizability across various network contexts and datasets.

The Voting Classifier has the highest true positive rate (TP = 1718), suggesting it is the best at correctly identifying normal activity.

9. Conclusion

The evaluation of three models; Support Vector Machine (SVM), K-nearest Neighbor (KNN), and a Hybrid model on insider's threat detection revealed distinct performance differences. While SVM demonstrated the weakest performance with the highest number of false positives (574), indicating a tendency to misclassify normal instances as threats, KNN exhibited a strong overall performance with balanced accuracy, precision, recall, and F1 score. Remarkably, the Hybrid model surpassed both individual models, showcasing exceptional accuracy (99%) and precision (99%), making it the most reliable choice for detecting insider's threats with minimal false positives and maximum effectiveness. These findings emphasize the importance of employing sophisticated model combinations for optimal insider's threat detection, thereby enhancing security measures within organizations.

Recommendations

Based on the study findings, it is recommended to implement the hybrid model, incorporating SVM, Voting Classifier, and KNN, in real-world cybersecurity systems to enhance insider's threat detection across diverse application areas. Emphasis should be placed on continuous monitoring, particularly in sectors like finance, critical infrastructure, and organizational networks. Additionally, customizing the hybrid model to align with specific industry requirements is crucial for optimal performance. Encouraging collaborative research efforts to further refine and optimize the hybrid model, as well as exploring seamless integration with existing cyber security frameworks, will contribute to its effective application and advancement in the field. These recommendations collectively aim to guide practical implementations and foster advancements in insider's threat detection across various sectors.

Compliance with ethical standards

Disclosure of conflict of interest

The authors report no declarations of interest.

References

[1] Rodriguez, A. and Smith, B. (2023). Enhancing Model Performance in Insider Threat Detection: The Role of Dynamic Feature Selection. Proceedings of the International Conference on Cybersecurity, 25, 245-258.

[2] Akazue, M. I. Debekeme, I. A. Edje, A. E. Asuai, C. Osame, U. J. (2023). Unmasking Fraudsters: Ensemble Features Selection to Enhance Random Forest Fraud Detection. Journal of Computing Theories and Applications, Vol 1, issue 2, pp 201-211

[3] Akazue, M. I. Clive, A. Abel, A.E, Omede, E, and Ufiofio, E. (2023b). Cybershield: Harnessing Ensemble Feature Selection Technique for Robust Distributed Denial of Service Attacks Detection, Kongzhi yu Juece/Control Decis., vol. 38, no. 3, 2023.

[4] Noever, D. (2019). Classifier Suites for Insider Threat Detection. arXiv:1901.10948. Available online: https://arxiv.org/abs/1901.10948.

[5] Ojugo, A. A. and Ekurume, E. (2021). Deep Learning Network Anomaly-Based Intrusion Detection Ensemble for Predictive Intelligence to Curb Malicious Connections: An Empirical Evidence, International Journal of Advanced Trends in Computer Science and Engineering, vol. 10, no. 3, pp. 2090–2102, Jun. 2021, doi: 10.30534/ijatcse/2021/851032021.

[6] Akazue, M.I. Nonum, I. Omede, E. and. Edje, A. E. (2023c) Application of A Multi-Layered Optimal Classifier for Telecommunication Fraud Prediction System, International Journal of Trend in Research and Development, Volume 10(4), 225-231

[7] Ajenaghughrure, I., Sujatha, P. Akazue, M.I. (2017). Fuzzy based multi-fever symptom classifier diagnosis model. International Journal of Information Technology and Computer Science, Vol. 9, No. 10, pp. 13–28. DOI: 10.5815/ijitcs.2017.10.02.

[8] Akazue, M. I. Onyeacholem, I. J., Omede, E (2024). Application of Supervised Machine Learning Algorithm with an Intrusion Detection System for Grazing Animals' Detection, FUPRE JOURNAL 8(1): 69-78

[9] Cappelli, D. Moore, A. Trzeciak, R. (2012). The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud). Addison-Wesley: Boston, MA, USA

[10] Akazue M. I. and Ajenaghughrure, I. B. (2015). Ant colony Optimization Algorithm Based Vehicle Theft Prediction-Prevention and Recovery System Model, International Journal of Innovative Research in Computer and Communication Engineering 3(9):8262-8277

[11] Ojugo A. A. Akazue, M. I. Ejeh, P. O. Odiakaose, C. C. Emordi, F. U. (2023a). DeGATraMoNN: Deep Learning Memetic Ensemble to Detect Spam Threats via a Content-Based Processing. Kongzhi yu Juece/Control and Decision, Volume 38, Issue 01, April, 2023, ISSN: 1001-0920

[12] Akazue, M. I. Ojugo, A. A. Yoro, R. E. Malasowe, B. O. Nwankwo, and O. (2022) Empirical Evidence of Phishing Menace Among Undergraduate Smartphone Users in Selected Universities in Nigeria, Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), vol. 28, no. 3, pp. 1756–1765, doi: 10.11591/ijeecs.v28.i3.pp1756-1765.

[13] Akazue, M. I. Okofu, S. N. Ojugo, A. A. P. O. Ejeh, Odiakaose, C. C. Emordi, F. U. Ako, R. E. Geteloma, V. O. (2024). Handling Transactional Data Features via Associative Rule Mining for Mobile Online Shopping Platforms, International Journal of Advanced Computer Science and Applications, Vol. 15, No. 3, pp 530-538

[14] Ojugo, A. A. and Eboka, A. O. (2020). Memetic Algorithm for Short Messaging Service Spam Filter Using Text Normalization and Semantic Approach, International Journal of Informatics and Communication Technology (IJ-ICT), vol. 9, no. 1, pp. 9–18. doi: 10.11591/ijict.v9i1.pp9-18.

[15] Ojugo, A. A. Akazue, M. I. Ejeh, P. O. Ashioba, N. C. Odiakaose, C. C. Ako, R. E. and Emordi, F. U. (2023b). Forging a User-Trust Hybrid Memetic Modular Neural Network Card Fraud Detection Ensemble: A Pilot Study, Journal of Computing Theories and Applications, vol. 1, no. 2, page 1-11

[16] Akazue, M. I. Ahweyevu, K. O. Ogeh, C. O. and Asuai, C. (2024). Development of a Real-time Phishing Detection Website via a Triumvirate of Information Retrieval, Natural Language Processing, and Machine Learning Modules, International Journal of Trend in Research and Development, Volume 11(1), pp 102 -108

[17] Chen, S. Li, J. and Wang, X. (2017). Anomaly Detection Algorithms for Insider Threat Identification. Journal of Information Security, 15(2), 78-92.

[18] Akazue, M . I (2015). A Survey of Ecommerce Transaction Fraud Prevention Models, proceedings of the International Conference on Digital Information Processing, Data Mining, and Wireless Communications, Dubai, UAE, 140-146

[19] Gavai, G. Sricharan, K. Gunning, D. Hanley, J. Singhal, M. Rolleston, R. (2015). Supervised and Unsupervised methods to detect Insider Threat from Enterprise Social and Online Activity Data. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. 2015, 6, 47–63. [CrossRef]

[20] Kulik, T. Dongol, B. Larsen, P.G. Macedo, H.D. Schneider, S. Tran-Jørgensen, P.W. Woodcock, J. (2022). A Survey of Practical Formal Methods for Security. Formal Aspects of Computing, 34(1), 1–39. DOI: 10.1007/s00165-021-00580-y. Available online: https://link.springer.com/article/10.1007/s00165-021-00580-y

[21] Rauf, U. Shehab, M. Qamar, N. Sameen, S. (2021). Formal Approach to Thwart Against Insider Attacks: A Bio-Inspired Auto-Resilient Policy Regulation framework. Future Generation Computer Systems, 117, 412–425. DOI: 10.1016/j.future.2020.10.019.

[22] Karev, D. McCubbin, C. Vaulin, R. (2017). Cyber Threat HuntingThrough the Use of an Isolation Forest. In Proceedings of the 18th International Conference on Computer Systems and Technologies, Ruse, Bulgaria, 23–24 June 2017; Association for Computing Machinery: New York, NY, USA, 2017; CompSysTech'17, pp. 163–170.

[23] Taylor, G. Anderson, H. and Roberts, F. (2021). Attention-based LSTM for Insider Threat Detection: Modeling Normal User Behavior." Journal of Machine Learning Research, 22(3), 567-580.

[24] Daniels, T. White, J. and Harris, M. (2023). Hybrid LSTM-XGBoost Model for Insider Threat Detection. Journal of Cybersecurity, 15(2), 210-224.

[25] Ojie, D. Akazue, M. I. and Imianvan, A. (2023). A Framework for Feature Selection using Data Value Metric and Genetic Algorithm, International Journal of Computer Applications, Volume 184, Issue 43, p14-21, doi:10.5120/ijca2023922533

[26] Hughes, N. Richardson, T. and Sanchez, E. (2023). CNN-Based Insider Threat Detection in Network Traffic Data. International Journal of Trend in Research and Development, 26(1), 210-224.

[27] Gavai, G. Sricharan, K. Gunning, D. Hanley, J. Singhal, M. Rolleston, R. (2015). Supervised and Unsupervised methods to detect Insider Threat from Enterprise Social and Online Activity Data. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. 2015, 6, 47–63. [CrossRef].

[28] Martinez, R., Johnson, K. and Garcia, L. (2021). Adversarial Training for LSTM-Based Insider Threat Detection. IEEE International Conference on Cyber Security and Resilience, 19(3), 567-580.

[29] Singh, M. Mehtre, B.M.; Sangeetha, S. (2019). User Behavior Profiling Using Ensemble Approach for Insider Threat Detection. In Proceedings of the 2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA), Hyderabad, India, 22–24 January 2019; pp. 1–8.

[30] Mega, O. G., Akazue, M. I., Apene, O. Z., & Hampo, J. A. (2024). Adoption of Blockchain Technology Framework for Addressing Counterfeit Drugs Circulation. European Journal of Medical and Health Research, 2(2), 182-196. https://doi.org/10.59324/ejmhr.2024.2(2).20

[31] Ojugo, A. A. Ejeh, P. O. Akazue, M. I. Ashioba, N. C. Odiakaose, C. C. Ako, R. E Nwozor, B. and Emordi, F. U. (2023c). CoSoGMIR: A Social Graph Contagion Diffusion Framework using the Movement-Interaction-Return Technique, Journal of Computing Theories and Applications, 1 (2) 163 – 173

[32] Parveen, P. Thuraisingham, B. (2012). Unsupervised Incremental Sequence Learning for Insider Threat Detection; Proceedings of the 2012 IEEE International Conference on Intelligence and Security Informatics; Washington, DC, USA. 11–14 June 2012; pp. 141–143.

[33] Koutsouvelis, V. Shiaeles, S. Ghita, B. Bendiab, G. (2020). Detection of Insider Threats using Artificial Intelligence and Visualisation. In Proceedings of the 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29 June–3 July 2020; pp. 437–443.

[34] Orizio, R. Vuppala, S. Basagiannis, S. Provan, G. (2022). Towards an Explainable Approach for Insider Threat Detection: Constraint Network Learning. In Proceedings of the 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), San Antonio, TX, USA, 5–7, pp. 42–49.

[35] Smith, J. Brown, A. and Garcia, M. (2018). Evaluating Machine Learning Models for Insider Threat Detection. Journal of Cybersecurity Research, 12(3), 45-58.

[36] Jones, D. and Wang, L. (2019). Integrating Diverse Data Sources for Insider Threat Detection. International Journal of Innovative Science and Research Technology, 25(4), 112-126.

[37] Garcia, E. Martinez, J and Kim, S. (2020). Explainable Machine Learning Models for Insider Threat Detection. International Journal of Trend in Research and Development,, 8(2), 75-89.

[38] Chen, S. Li, J. and Wang, X. (2017). Anomaly Detection Algorithms for Insider Threat Identification. Journal of Information Security, 15(2), 78-92.

[39] Wong, T. and Patel, R. (2019). Harnessing Diverse Data Sources for Insider Threat Detection. International Journal of Cybersecurity, 5(3), 112-126.

[40] Gupta, A. Singh, R. and Kim, Y. (2020). Advancements in Explainable Machine Learning Models for Insider Threat Detection. International Journal of Cybersecurity Networks, 18(4), 75-89.

[41] Smith, A. Brown, B. and Johnson, C. (2019). Unveiling Insider Threats: Anomaly Detection in Employee Communications. Journal of Cybersecurity, 10(3), 123-137.

[42] Li, Q. and Patel, A. (2022). Exfiltration Pathway Analysis for Insider Threat Detection: A Network-Centric Approach. Journal of Cybersecurity Networks and Infrastructures, 13(4), 120-138.