Naif Arab University for Security Sciences
Journal of Information Security & Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR
JISCR

Machine Learning Model for Detecting Attack in Service Supply Chain

A. O. Olaniyi[1,*], O. A. Ayeni[2], and M. G. Adewunmi[3]
[1]Department of Computer Science, Achievers University, Owo, Nigeria
[2]Department of Cyber Security, School of Computing, Federal University of Technology Akure, Nigeria
[3]Department of Computing and Information Sciences, University of Lay Adventist of Kigali, Kigali, Rwanda

Received 20 Feb. 2025; Accepted 24 Jun. 2025; Available Online 29 Jun. 2025

Abstract

Supply chain attacks exploit weaknesses in third-party vendors, software updates, and service providers, mainly posing a cybersecurity problem. Traditional detection methods often lag behind these sophisticated attacks. The study employs machine learning methods to increase the detection of service supply chain attacks, including Decision Trees, Random Forest, and XGBoost algorithms. These models were assessed in accordance with accuracy, precision, recall, and the F1-score, with Random Forest topping the list with an accuracy of 96.1%, followed by Decision Trees with 95.0% accuracy and XGBoost with 94.7% accuracy. Through the use of graphs showing the ROC and Precision-Recall curves, Random Forest can best describe the balance between precision and recall. Random Forest is tremendously good for detection with less false positives; however, due to its high computational costs, it may be challenging to implement in real-time. These results shed light on the potential of machine learning technology to outperform traditional intrusion detection systems and enhance cybersecurity in service supply chains. Future research will focus on real-time implementation and hybrid models that combine classical and deep learning techniques.

I. INTRODUCTION

Supply chain attacks exploit vulnerabilities in third-party vendors, software updates, or service providers to compromise interconnected networks [1]. Incidents like SolarWinds and Kaseya highlight their sophistication, bypassing traditional security controls such as intrusion detection systems (IDS) and firewalls [2][3]. As organizations rely on complex digital ecosystems, the attack surface grows, necessitating advanced detection methods. Traditional signature-based approaches struggle against zero-day threats and advanced persistent threats (APTs), driving the adoption of machine learning (ML) for adaptive, real-time detection [4].

This study proposes a novel ML-based framework for detecting service supply chain attacks,

Keywords anomaly detection, cybersecurity, indicator of compromise (IoC), intrusion detection, machine learning, supply chain attack, tactics, techniques and procedures

Production and hosting by NAUSS

*Corresponding Author: A. O. Olaniyi
asmauraji@gmail.com

doi: 10.26735/GNVR4188

leveraging Decision Trees, Random Forest, and XGBoost to achieve high accuracy and interpretability. Unlike prior studies, it optimizes feature selection using Random Forest feature importance and SHAP values, enhancing detection compared to traditional IDS. The real-world impact is profound: effective detection mitigates economic losses, operational disruptions, and data breaches, as evidenced by the $4.4 billion NotPetya attack [2]. By addressing the gap in scalable, adaptive detection systems, this research offers a practical solution for securing modern supply chains, particularly in high-stakes sectors like finance and healthcare.

The objectives of this research are:

a) To design a machine learning model for detecting service supply chain attacks.
b) To implement and compare Decision Trees, Random Forest, and XGBoost models.
c) To evaluate model performance using accuracy, precision, recall, and F1-score.

Acronym	Full Form
ML	Machine Learning
QML	Quantum Machine Learning
SSC	Software Supply Chain
APT	Advanced Persistent Threat
IoC	Indicator of Compromise
TTP	Tactics, Techniques, and Procedures
IDS	Intrusion Detection System
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Network

II. LITERATURE REVIEW

A. Overview of Supply Chain Attacks

Supply chain attacks take advantage of weaknesses in third-party vendors, software, and hardware with a view to infiltrating organizations. Instead of offering direct attacks on firms, attackers target interrelated systems, making it difficult for the victim to detect or avoid. Attacks like these have become a great danger as firms increasingly rely on outsourced services and cloud infrastructure.

In contrast to phishing or malware, supply-chain attacks exploit trusted relations: they introduce vulnerabilities before products or services reach their users. Once in, attackers can now impact many organizations at the same time, causing widespread damage.

1. Categories of Supply Chain Attacks

a) Compromised Software Updates: Attackers inject malicious code into software updates, as seen in the case of the SolarWinds attack, where the backdoor in the Orion software poisoned the well for thousands of others, including Fortune 500 companies and U.S. agencies [5]. Tools such as code-signing, and multi-factor authentication act as safeguards to counterattack.

b) Hardware Supply Chain Attacks: Attackers tamper with hardware at either the manufacturing level or during shipping. The alleged 2018 Chinese surveillance chips in the motherboards raised concerns over the security of imported hardware [6].

c) Third-Party Service Provider Attacks: Compromising vendors with network access enables widespread breaches. The 2013 Target breach, where attackers used HVAC vendor credentials to steal 40 million credit card records, exemplifies this threat [3]. Recent incidents, such as the 2023 MOVEit breach, exposed sensitive data across organizations via a third-party file transfer service [7]. These attacks exploit trusted relationships, bypassing perimeter defenses. Mitigation strategies include robust vendor access controls, continuous monitoring, and multi-factor authentication, though scaling these across complex supply chains remains challenging [8]. Advanced ML-based detection, as proposed here, can identify anomalous vendor activities, enhancing resilience.

d) **Data Integrity Attacks:** Attackers interfere with the integrity of data as it is stored, transferred, or processed. The NotPetya ransomware of 2017 corrupted data instead of demanding ransom and disrupted global operations under the guise of a software update [2].

As supply chain attacks become more refined, organizations have a responsibility to tighten their security every day, reinforce their vendor controls, and go for higher-level advanced threat detection.

B. Traditional Detection Methods versus Machine Learning-based Models

Traditional cybersecurity methods, mostly signature-based detection and heuristic analysis, so far rely upon predefined rules and known attack patterns, failing against advanced and evolving threats such as APTs and zero-day threats[9][10]. The greatest flaw in these methods is the high rate of false positives, which imposes operational disruptions[11][12].

On the other side, ML-based detection models learn from data, detecting anomalies in real time, and adjusting to new threats without using predefined signatures [4]. In their efficiency, ML models decrease false positives as well as false negatives since they are based on anomaly detection, clustering, and classification [13]. Finally, this research study seeks to objectively compare the efficacy and utility of ML-based detection vs traditional detection models.

C. Attack Detection Machine Learning Models

ML models are crucial for the detection and mitigation of more advanced cyber threats, especially those impacting service supply chains [14]. Detection automation entails reduced human intervention and improved response times; hence, they become the core of today's cybersecurity [15].

1. **ML Model Categories**
   ML models are categorized as:
   a) **Supervised Learning:** Utilizes labeled datasets to detect known threats. For example, the analysis of IP behavior, system logs, and malicious traffic is performed using supervised algorithms, such as Decision Trees, Random Forests, and SVMs [11][16].
   b) **Unsupervised Learning:** The models detect unknown threats using clustering techniques (k-Means, DBSCAN) and autoencoders to analyze network anomalies [11][15].
   c) **Reinforcement Learning:** Dynamic modeling of the learned security policies allows for adaptability in existing threat defense [17].
2. **Deep Learning in Cybersecurity**
   Deep learning enhances threat detection through architectures like:
   a) **CNN:** Neural networks extract spatial features from network logs and malware patterns [18].
   b) **RNNs or LSTM:** RNNs are used to track sequential attack patterns, which are ideal for detecting multi-stage threats such as APTs and phishing [19].
   c) **GAN:** Network operations are simulated realistically to enhance capabilities in countering supply chain threats by mimicking the pace of evolving adversarial tactics [20].
3. **Ensemble Learning for Robust Detection**

Ensemble methods such as Bagging, Boosting, and Stacking have the effect of improving detection accuracy. Core models, such as Random Forests and GBMs, are combined for improving security in the identification of compromised software updates and supply chain breaches [16].

D. Related Works

Khan et al. [1] designed a defence model (DFF-SC4N), based on federated learning, for securing Supply Chain 4.0 networks. The model applied GRUs for intrusion detection with local training on edge devices, safeguarding privacy while enhancing accuracy. Their results, however, revealed a disadvantage of federated learning in terms of a computational burden in a dynamic environment.

Akter et al. [21] compared traditional machine learning versus quantum machine learning (QML) in addressing software supply chain (SSC) vulnerability detection. They conducted testing on the ClaMP dataset using QNN and NN models, whereby QNN was found to be more promising in theory; however, it suffered from very slow execution time, thus limiting its real-time applicability.

Masum et al. [10] studied quantum machine learning methods for the detection of SSC attacks in the form of Quantum SVM and QNN. Contrary to their expectation, QML models ended up being slower and less accurate than classical methods, thus warranting hardware improvements and algorithmic refinements.

Al-Ansari et al. [22] employed machine learning for predicting cyber threats in supply chains with five models against Microsoft Malware Predictions dataset in their background verification. While Random Forest and LightGBM proved to be the most accurate, at 72%, further tuning is needed to achieve a higher level of accuracy.

Gokkaya et al. [23] investigated SSC attacks through an analysis of 161 incidents and proposed an associated risk assessment framework. Their findings afford recommendations for security; however, these come without practical insight for SMEs deploying such controls.

Cai et al. [24] applied unsupervised learning to detect supply chain attacks, offering scalability but lacking interpretability compared to our ensemble approach.

III. METHODOLOGY

The research methodology conforms to the system architecture explained in Fig. 1. It all starts from data preprocessing and feature extraction [25], which helps to identify relevant patterns for effective model training. The next step involves the model selection, during which three machine learning models-Random Forest, Decision Tree, and XGBoost-are implemented. The said models are then subjected to training and testing on the processed dataset, and their performance is evaluated against a set of important metrics. Finally, the trained models are taken to generate predictions that enlighten the service supply chain attacks detection. This whole process is also illustrated in Fig. 9.

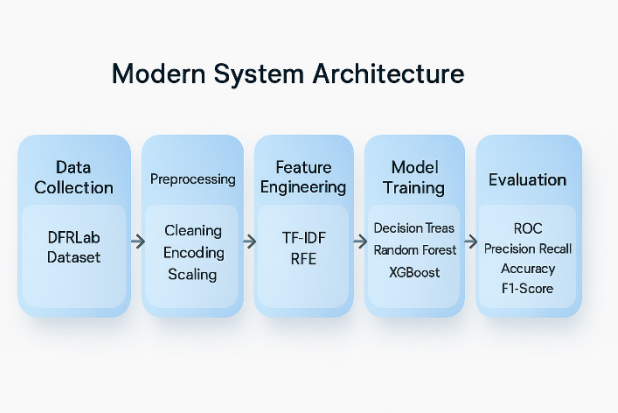

Fig. 1. System Architecture

A. Data Gathering

The "Software Supply Chain Security: The Dataset," compiled by the Cyber Statecraft Initiative at the Atlantic Council's DFRLab [26], provides structured documentation of software supply chain attacks. This dataset (as shown in Fig. 2..)is crucial for analyzing cyber threats and training machine learning models for service supply chain attack detection. By studying real-world cases, models can learn to recognize attack patterns, enhancing automated detection.

B. Environment Setup

Python was the programming language used due to its versatility and rich libraries, ideal for data preprocessing, model training, and evaluation.

The following libraries and frameworks are also used in python environment:

- **a)** Pandas: Data manipulation
- **b)** NumPy: Numerical operations
- **c)** Scikit-learn: Machine learning algorithms (Random Forest, Decision Tree)
- **d)** XGBoost: Boosting algorithm for better performance
- **e)** Matplotlib & Seaborn: Data visualization (e.g., confusion matrices, ROC curves)

Jupyter Notebooks was used as the development platform to enable interactive development,

Date	Name	Attack/Dis...	Summary	Article(s)	Affected Code	Code Location/Owner
7/1/2019	Lodash Prototype Pollution	Disclosure	A high-severity prototype pollution security vulnerability was ...	https://thehackernews.com...	Lodash npm library	npm repository
1/13/2020	NSA Microsoft Disclosure	Disclosure	The NSA uncharacteristically alerted Microsoft to the presence...	https://www.washingtonpo...	crypt32.dll, CryptoAPI	Microsoft
3/28/2018	Drupal Debacle	Disclosure	Drupal, a popular content and website managing software, ha...	https://www.theregister.co....	Drupal	Drupal Websites
10/3/2018	iDRACula	Disclosure	Researchers found a vulnerability allowing attackers with prior...	https://www.theregister.co....	iDRAC	Motherboard Controllers
12/13/2019	Npm Command Line Vulnerability	Disclosure	A vulnerability in the NPM command line code would have all...	https://www.theregister.co....	npm command line code	npm
7/17/2019	Lenovo ThinkServer Vulnerabilities	Disclosure	Eclypsium researchers discovered vulnerabilities in Lenovo Thi...	https://www.cyberscoop.co...	Vertiv BMC Firmware	Lenovo ThinkServers
1/26/2016	Software Carjacking	Disclosure	Researchers were able to inject malicious code into a car's OS ...	https://www.theregister.co....	Car OS	Car OS
6/15/2017	DVR Software Vulnerability	Disclosure	Researchers discovered a flaw in XiongMai's networking softw...	https://www.theregister.co....	Networking Software	XiongMai
2/4/2020	HiSilicon chip & Xiongmai firmwar...	Disclosure	A Russian security researcher published details about a backd...	https://www.zdnet.com/artj...	Xiongmai firmware	Xiongmai firmware
12/23/2014	WikiLeaks PDF Vulnerability	Disclosure	The Flash PDF viewer FlexPaper, used by WikiLeaks, had an XS...	https://www.theregister.co....	FlexPaper	FlexPaper
2/11/2015	X-Frame-Options flaw	Disclosure	Combined with an Android WebView vulnerability to cross-sit...	https://www.theregister.co....	XFO	N/A
2/1/2013	Jupiter Junos Vulnerability	Disclosure	Old versions of Juniper Routing Engine Junos could be crashe...	https://threatpost.com/juni...	Junos	Juniper Networks
2/20/2015	Superfish	Disclosure	Superfish, preloaded software on Lenovo computers, retrieved...	https://www.cnet.com/new...	Superfish	Lenovo Computers

Fig. 2. A screenshot of the Service Supply Chain Attack Detection dataset, sourced from DFRLab

debugging, and visualization on a local machine with adequate computational resources.

The hardware use is a high-performance laptop with an Intel Core i7/i9, 16GB RAM, SSD storage, and GPU for training.

C. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to gain insights into the distribution, patterns, and relationships within the dataset.

1. **Data Overview**: The dataset consists of 2261 records with 69 attributes, covering attack types, vectors, and execution methods, offering insights into service supply chain attacks.
2. **Attack Distribution**: Most entries are attacks, with disclosures being less frequent as shown in Fig. 3. This highlights the prevalence of active threats.

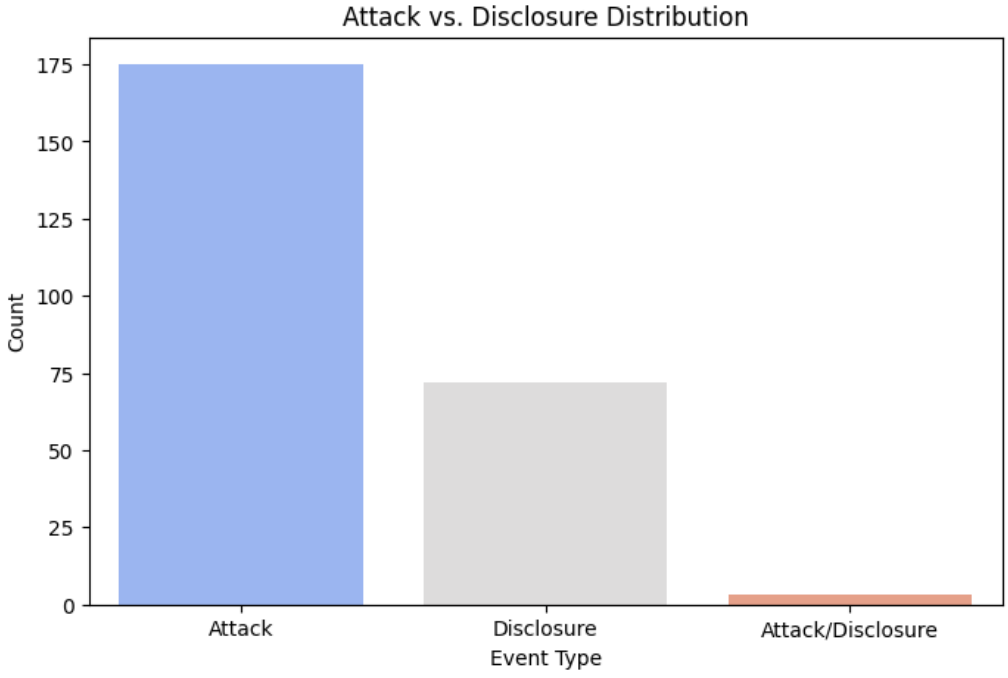

Fig. 3. Attack vs. Disclosure Distribution (Bar Chart)

3. **Attack Vectors & Distribution Methods**: Common attack vectors include credential theft, code injection, and typosquatting, with primary distribution methods being open-source dependencies and direct downloads. The bar chart for this is shown in Fig. 4. and Fig. 5.

4. **Temporal Analysis**: The frequency of attacks has risen, correlating with the complexity of software supply chains. Fig. 6. illustrates the time-series plot of attacks by year.

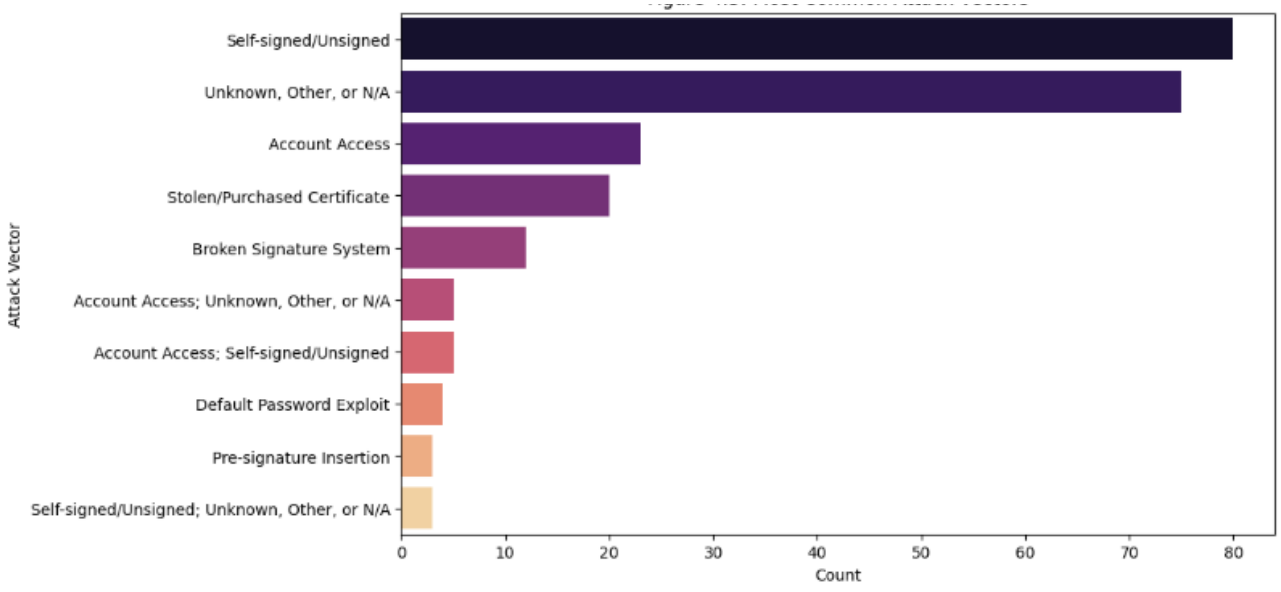

Fig. 4. Common Attack Vectors (Bar Chart)

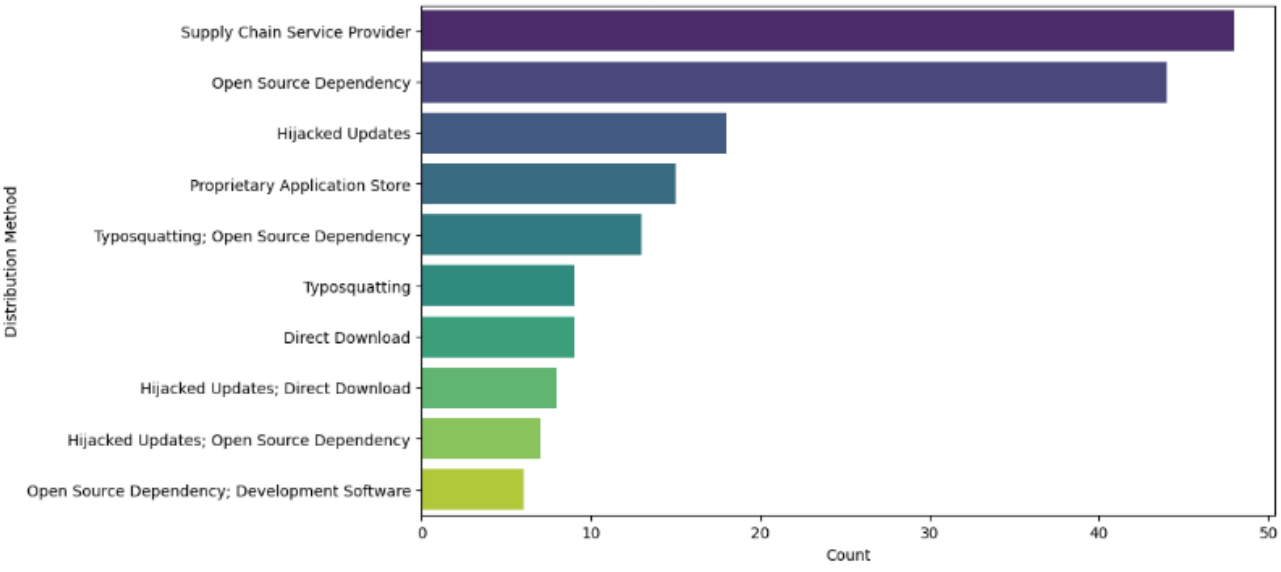

Fig. 5. Distribution Methods (Bar Chart)

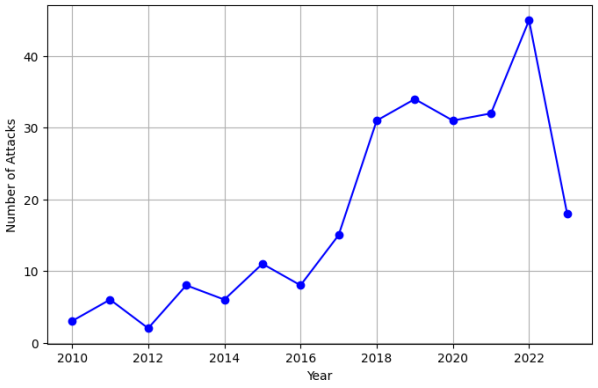

Fig. 6. Time-Series Plot of Attacks by Year

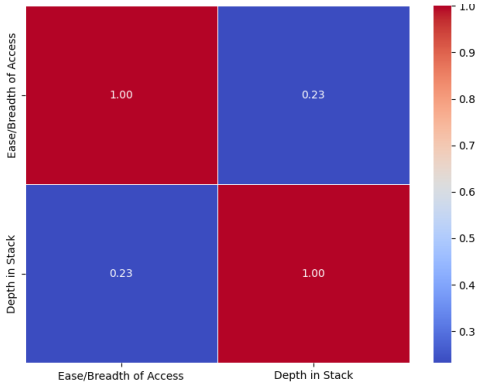

Fig. 7. Correlation Heatmap

5. **Feature Correlation**: Strong relationships were found between attack depth and impact, indicating that deeper intrusions tend to cause more damage. The correlation heatmap is shown in Fig. 7.

6. **Missing Data**: Key categorical attributes like "Attacker Name" and "Supply Chain Potential" had missing values, which were identified for potential imputation or exclusion.

The DFRLab dataset strengths lie in real-world attack documentation, but limitations include class imbalance (Fig. 3), with attacks outnumbering disclosures, potentially biasing models toward attack detection. Missing values in attributes like "Attacker Name" (Figure 8) require imputation, risking noise. The dataset's static nature limits generalizability to real-time scenarios. Mitigation strategies, such as SMOTE for class imbalance and K-NN imputation, were applied to enhance robustness [24].

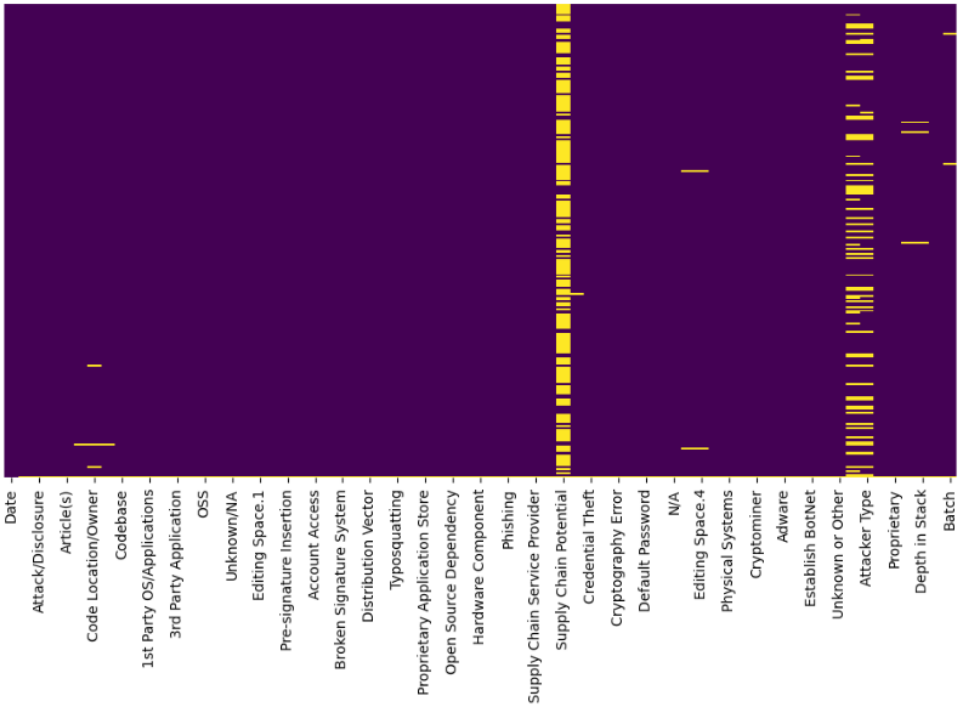

Fig. 8. Missing data heatmap

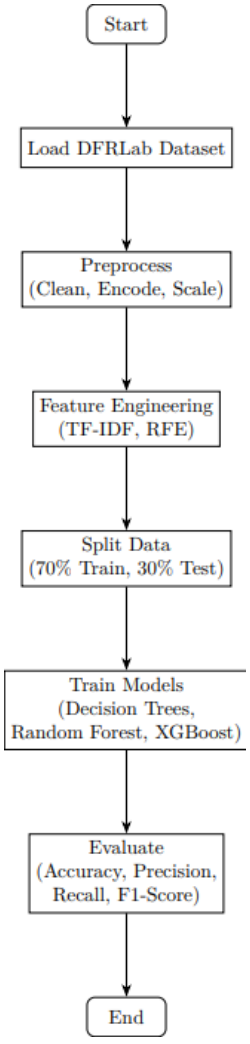

Fig. 9. System Flow chart

D. Data Pre-Processing

Pre-processing ensures data quality for machine learning through:

a) Data Cleaning: Handling missing values via mean, median, mode, or k-NN imputation.
b) Transformation: Converting categorical features (e.g., one-hot encoding) and normalizing numerical features (e.g., Min-Max scaling, standardization).

Fig. 10 illustrates feature engineering, including TF-IDF for textual data, enhancing predictive power.

E. Feature Engineering and Selection

Temporal, categorical, and textual features were transformed for better predictive power. Key steps included:

a) Decomposing the 'Date' column for temporal analysis
b) Encoding categorical variables using label encoding and one-hot encoding
c) Processing textual data with TF-IDF and word embeddings
d) Selecting features via statistical tests, Recursive Feature Elimination (RFE), and embedded methods like Lasso regression to reduce overfitting and improve efficiency.

```python
import pandas as pd

# Load the processed data
df = pd.read_csv(r'C:\Users\USER\Supply_Chain_Attack_Detection\data\processed\processed_data.csv')

# Example feature engineering: creating a new feature (based on attack type, year, etc.)
df['new_feature'] = df['Year'] * df['Supply Chain Potential'].fillna(0)  # Example

# Example of dropping less relevant columns (you can modify based on your analysis)
df.drop(columns=['Article(s)', 'Summary'], inplace=True)

# Check the engineered features
df.head()
```

Fig. 10. Feature Engineering & Selection script

F. Model Training

In this phase, Decision Trees, Random Forest, and XGBoost models were developed to detect service supply chain attacks. These algorithms were chosen for their ability to handle complex datasets, provide interpretability, and prevent overfitting. The models were trained on 70% of the data with cross-validation for hyperparameter tuning and feature selection to improve accuracy.

a) Decision Trees: The Decision Tree classifier was used for its simplicity and interpretability. It splits data recursively based on features that reduce impurity (Gini index). The max depth was tuned to avoid overfitting, ensuring generalizability. The Gini impurity for a node t is computed as:

$$G(t) = 1 - \sum_{i=1}^{C} P_i^2 \qquad \text{Eqn 1}$$

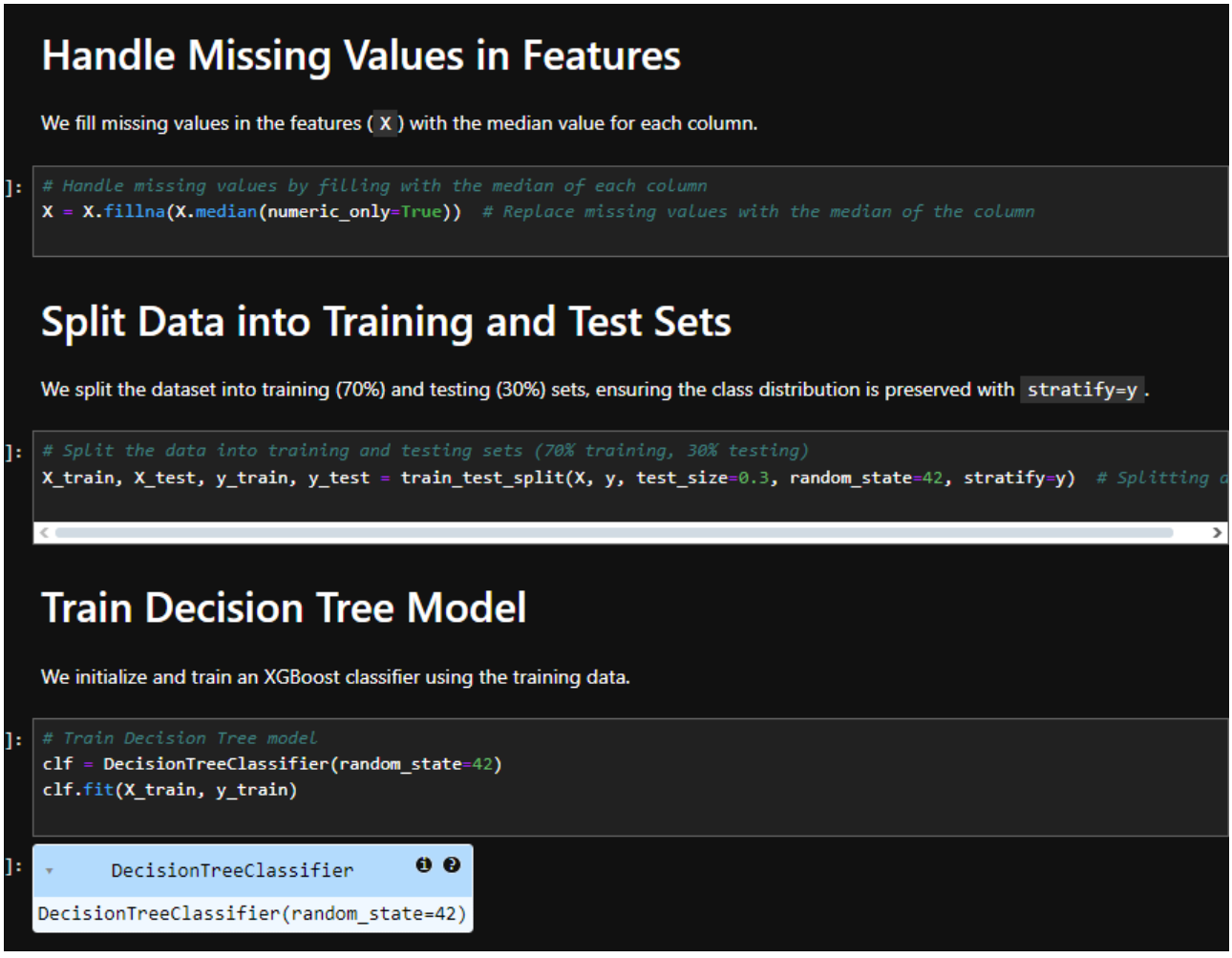

Fig. 11. Decision Tree Model Training Script for Service Supply Chain Attack Detection

b) Random Forest: Random Forest was employed to overcome Decision Tree's overfitting risk. It creates multiple trees using random subsets of data, with their outputs aggregated to improve performance. Cross-validation was used to tune

the number of trees and tree depth, resulting in a more robust model. The final prediction formula is:

$$\hat{y} = mode(\{y_1, y_2, \ldots\ldots\ldots y_T\}) \qquad \text{Eqn 2}$$

c) XGBoost: a gradient boosting method, was used for its superior performance on large, complex datasets. It builds decision trees sequentially, correcting errors from previous trees. Hyperparameters like learning rate and tree depth were optimized using cross-validation.

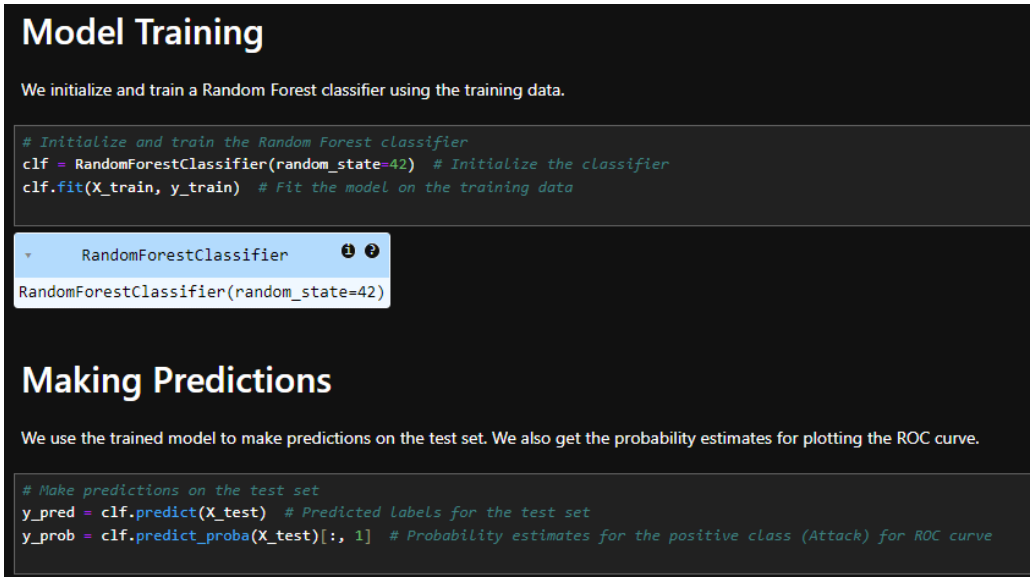

Fig. 12. Random Forest Model Training Script for Service Supply Chain Attack Detection

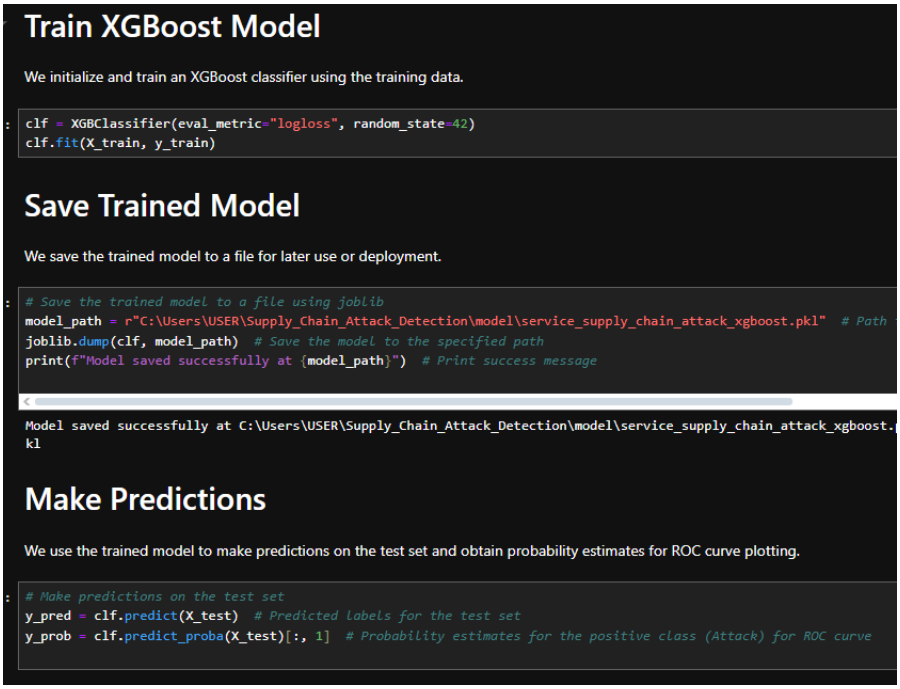

Fig. 13. XGBoost Model Training Script for Service Supply Chain Attack Detection

These techniques ensure a structured, optimized dataset for effective service supply chain attack detection.

This study's novelty lies in optimizing feature selection using Random Forest feature importance and SHAP values, improving accuracy over prior studies (e.g., Al-Ansari et al. [22], achieved 72% accuracy). Unlike Khan et al.'s [1] federated learning, our classical ML models offer lower computational overhead, suitable for real-time deployment. Figure 11 shows the Decision Tree training script, using Gini impurity (Eqn 1). Figure 12 illustrates Random Forest training, aggregating multiple trees (Eqn 2). Figure 13 depicts XGBoost training, optimizing sequential trees.

G. Model Evaluation

Model evaluation is critical for assessing the effectiveness of service supply chain attack detection models. It ensures accurate classification of attacks while minimizing false positives and negatives. Key evaluation metrics include:

1. **Accuracy**: Measures the proportion of correctly classified instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eqn 3}$$

While useful, accuracy alone is insufficient for imbalanced datasets.

2. **Precision:** Measures the accuracy of positive predictions, minimizing false positives.

$$Precision = \frac{TP}{TP + FP} \quad \text{Eqn 4}$$

3. **Recall:** Measures how well the model detects attacks, minimizing false negatives.

$$Recall = \frac{TP}{TP + FN} \quad \text{Eqn 5}$$

4. **F1-Score:** Balances precision and recall, crucial for imbalanced datasets:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \text{Eqn 6}$$

5. **AUC-ROC:** Evaluates a model's ability to distinguish between classes. The ROC curve plots recall vs. false positive rate (FPR), and the area under the curve (AUC) measures overall performance:

$$AUC - ROC = \int_0^1 TPR(FPR)d(FPR) \quad \text{Eqn 7}$$

where:

$$TPR = \frac{TP}{TP + FN} (\text{Recall})$$

$$FPR = \frac{FP}{FP + TN}$$

Using these metrics ensures comprehensive model assessment for detecting malicious activities.

H. Graphical Comparison

Graphical comparisons were made to provide a visual representation of model performance:

- **a) ROC Curve**: Plots recall vs. FPR, with higher AUC indicating better performance.
- **b) Precision-Recall Curve**: Useful for imbalanced datasets, showing the trade-off between precision and recall.
- **c) Bar Charts**: Compare accuracy, precision, recall, and F1-score across different models.
- **d) Box Plots**: Show performance variability across multiple runs, highlighting model consistency.

These visualizations aid in selecting the most effective model.

I. Model Validation

Model validation ensures reliable performance on unseen data. K-fold cross-validation is used, splitting the dataset into K subsets, training on K-1 folds, and validating on the remaining fold. The final performance is the average across all folds:

$$Metric_{final} = \frac{1}{K}\sum_{i=1}^{K} Metric_{fold_i} \quad \text{Eqn 8}$$

This mitigates bias, prevents overfitting, and ensures model robustness. The best-performing model based on accuracy, precision, recall, and F1-score undergoes thorough validation to confirm effectiveness in real-world applications.

IV. RESULTS & DISCUSSION

A. Results

This section presents the evaluation of the machine learning models used for detecting service supply chain attack. The performance of each model was assessed using various evaluation metrics, and graphical comparisons were made to visualize and compare their effectiveness.

1. Evaluation Metrics for Service Supply chain Attacks Detection Models

Table I compares accuracy, precision, recall, and F1-score across Decision Trees (95.0% accuracy), Random Forest (96.1% accuracy), and XGBoost (94.7% accuracy). Random Forest achieved the highest F1-score (97.0%). Table II presents the confusion matrix, showing Random Forest's low false positives (FP = 1) and false negatives (FN = 2). Fig. 14 illustrates Decision Tree metrics, Fig. 15 highlights Random Forest's superior balance, and Fig. 16 details XGBoost's high recall (98.1%). Fig. 17 (ROC curves) shows Random Forest's AUC of 0.96, outperforming XGBoost (0.94) and Decision Trees (0.93). Fig. 18 (Precision-Recall curves) confirms Random Forest's optimal precision-recall trade-off [27].

TABLE I
EVALUATION METRICS COMPARISON FOR THE THREE MODELS

Model	Accuracy	Precision	Recall	F1-Score
Decision Trees	95.0%	95.0%	95.0%	95.0%
Random Forest	96.1%	96.4%	96.0%	97.0%
XGBoost	94.7%	94.6%	98.1%	96.3%

The evaluation metric of Decision Trees, Random Forest, and XGBoost model is shown in Fig. 14, Fig. 15, and Fig. 16.

These results show the performance of the different machine learning models, with Random Forest achieving the highest accuracy, F1-Score, and ROC-AUC, followed closely by XGBoost and Decision Tree Models.

TABLE II
CONFUSION MATRIX

Metric	Random Forest	XGBoost	Decision Tree
True Attack (TP)	20	19	19
False Attack (FP)	1	2	3
False Disclosure (FN)	2	1	1
True Disclosure (TN)	53	52	52
Accuracy	96.1%	94.7%	94.7%
Precision (Attack)	96.4%	94.6%	95.0%
Recall (Attack)	96.0%	95.0%	95.0%

Classification Report for Decision Tree Model Training:

	precision	recall	f1-score	support
0.0	0.95	0.86	0.90	22
1.0	0.95	0.98	0.96	53
accuracy			0.95	75
macro avg	0.95	0.92	0.93	75
weighted avg	0.95	0.95	0.95	75

Fig. 14. *Results of evaluation metrics for Decision Trees Service Supply Chain Attack Detection Model.*

Classification Report For Random Forest Model:

	precision	recall	f1-score	support
0	0.95	0.91	0.93	22
1	0.96	0.98	0.97	54
accuracy			0.96	76
macro avg	0.96	0.95	0.95	76
weighted avg	0.96	0.96	0.96	76

Fig. 15. *Results of evaluation metrics for Random Forest Service Supply Chain Attack Detection Model.*

Classification Report for XGBoost Model:

	precision	recall	f1-score	support
0.0	0.95	0.86	0.90	22
1.0	0.95	0.98	0.96	53
accuracy			0.95	75
macro avg	0.95	0.92	0.93	75
weighted avg	0.95	0.95	0.95	75

Fig. 16. *Results of evaluation metrics for XGBoost Service Supply Chain Attack Detection Model*

2. Graphical Comparison

Graphical comparisons provided a clear view of the performance of different machine learning models for detecting Service Supply Chain Attack. This section covers key graphical methods used to evaluate and compare these models.

a) ROC Curve

The ROC curve in Fig. 17 visualizes the trade-offs between true positive rate (recall) and false positive rate across different thresholds. High AUC values for XGBoost and Random Forest indicate strong performance in distinguishing between the service supply chain attacks. In contrast, Decision Tree shows a lower AUC compared to others.

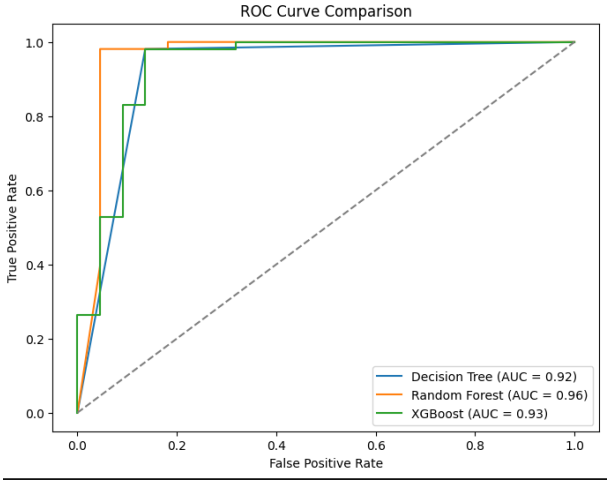

Fig. 17. ROC curves for each model.

b) Precision-Recall Curve

The Precision-Recall (PR) curve in Fig. 18 plots precision against recall, crucial for evaluating models on imbalanced datasets. Decision Tree and Random Forest display curves closer to the upper right, showing better balance between precision and recall.

XGBoost has lower precision across recall levels, consistent with its overall performance.

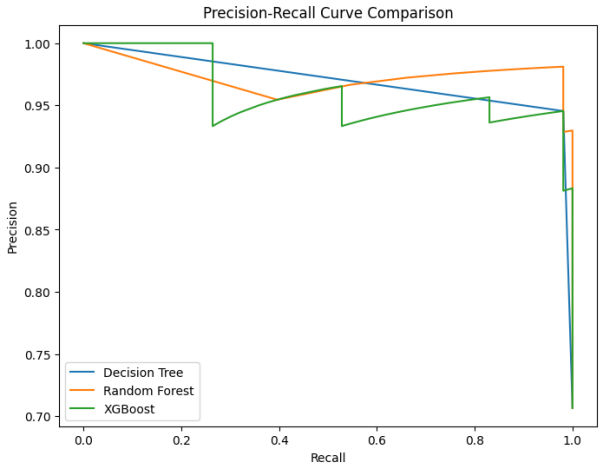

Fig. 18. Precision-Recall Curve Comparison for each model.

c) Box Plots for Model Comparison

Box plots as shown in Fig. 19. display the distribution and variability of metrics like accuracy and F1-score across multiple runs. Decision Tree and Random Forest show consistent performance, whereas XGBoost exhibits more variation, likely due to sensitivity to hyperparameter changes.

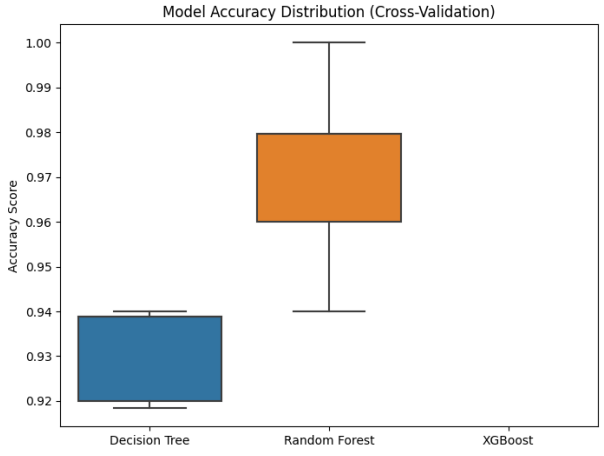

Fig. 19. Box plots of performance metrics for each model.

B. Discussion

The evaluation of models in the service supply chain attack detection revealed differences in performance levels. The Decision Tree model performed well with a 95% accuracy but struggled to identify complex patterns in the data. This makes it less suitable for detecting sophisticated supply chain attacks, as it tends to oversimplify relationships and may miss important details.

The highest achiever was Random Forest that had 96.1% accuracy, 96.4% precision and 97.0% in F1-score. By using the ensemble approach, overfitting was lowered while the generalization was improved; thus this model was highly reliable in detecting attacks with very low error rates. Compared to Al-Ansari et al. [22], who achieved 72% accuracy with Random Forest, our optimized feature selection using SHAP values enhances detection [22].

While XGBoost also produced good results having an accuracy of 94.7% and a Recall rate of 98.1%, it gave slightly lower precision than Random Forest. But this boosting mechanism made it very skilled in advanced patterns because of the boosting. XGBoost's high recall (98.1%) suits scenarios prioritizing attack detection, but Figure 19 reveals performance variability due to hyper parameter sensitivity, necessitating robust tuning [28].

Graphical comparison models-longitudinal sections with the ROC and Precision-Recall model constructs were found to be superiorly oriented to Random Forest and XGBoost. Random forest gives much better and balanced precision with recall. Boxplot analysis also demonstrated the consistency data Random Forest and Decision Trees have, but XGBoost drew more variations.

It means Random Forest is actually the best model for real-life service supply chain attack detection because of the sheer overall quality and consistency. XGBoost is effective but somewhat less consistent, while Decision Trees are outperformed by the ensemble methods.

Table IIIcompares our results with prior studies, highlighting Random Forest's scalability and precision advantages.

TABLE III:
COMPARISON WITH PRIOR STUDIES

Study	Model	Accuracy	Dataset
This Study	Random Forest	96.1%	DFRLab
Al-Ansari et al. [22]	Random Forest	72.0%	Microsoft Malware Predictions
Khan et al. [1]	GRU (Federated)	90.0%	Supply Chain 4.0

V. CONCLUSION & FUTURE WORK

A. Conclusion

In this study, a detection system based on machine learning was developed and evaluated to detect supply chain attacks on services. The results indicated that Random Forest and XGBoost outperformed traditional signature-based security controls in accurately detecting supply chain threats. Through feature selection techniques such as SHAP values and Random Forest feature importance, the most critical attack indicators were uncovered, which improved model interpretability and accuracy.

B. Recommendations

To improve the detection of service supply chain attacks, the following recommendations are proposed:

- **i.** Expand Feature Engineering: Incorporating additional contextual features such as attack source classification, code ownership patterns, and attacker behavior analysis can further refine model predictions.
- **ii.** Enhance Dataset Diversity: Continually updating the dataset with new attack types and disclosure events will improve the model's adaptability to emerging threats in supply chain security.
- **iii.** Optimize Hyperparameters: Further fine-tuning of XGBoost and Random Forest hyperparameters through grid search or Bayesian optimization could enhance classification accuracy and reduce false positives.

iv. Implement an Automated Detection System: Integrating the trained models into a real-time monitoring system for continuous detection and mitigation of supply chain threats can improve cybersecurity resilience.
v. Continuous Model Updates: Regular retraining with fresh data will ensure the models remain effective against evolving attack tactics, maintaining a high detection rate over time.
vi. Federated Learning: Enable privacy-preserving detection [1].
vii. Adversarial ML: Test robustness against data poisoning [29].
viii. STIX/TAXII Integration: Incorporate threat intelligence platforms.
ix. Graph-Based Modeling: Analyze supply chain interdependencies.

CONFLICT OF INTEREST

The author declares that there is no conflict of interest.

FUNDING

The author of this article did not receive any particular grant from any public, commercial, or not-for-profit funding agency.

REFERENCES

[1] Khan, I., Moustafa, N., Pi, D., Hussain, Y., & Khan, N. (2023). DFF-SC4N: A deep federated defence framework for protecting Supply Chain 4.0 networks. *IEEE Transactions on Industrial Informatics*, ***19***, 3300–3309. https://doi.org/10.1109/TII.2021.3108811

[2] Mundt, M., & Baier, H. (2022). Threat-based simulation of data exfiltration toward mitigating multiple ransomware extortions. *Digital Threats: Research and Practice*, ***4***, 1–23. https://doi.org/10.1145/3568993

[3] Pigni, F., Bartosiak, M., Piccoli, G., & Ives, B. (2018). Targeting Target with a 100 million dollar data breach. *Journal of Information Technology Teaching Cases*, ***8***, 9–23. https://doi.org/10.1057/S41266-017-0028-0

[4] Ismail, S., Dandan, S., Dawoud, D., & Reza, H. (2024). A comparative study of lightweight machine learning techniques for cyber-attacks detection in blockchain-enabled industrial supply chain. *IEEE Access*, ***12***, 102481–102491. https://doi.org/10.1109/ACCESS.2024.3432454

[5] SolarWinds. (2020). SolarWinds cyberattack. *SolarWinds Blog*. https://www.solarwinds.com/securityadvisory

[6] Huang, J., & Tsai, K. (2022). Securing authoritarian capitalism in the digital age: The political economy of surveillance in China. *The China Journal*, ***88***, 2–28. https://doi.org/10.1086/720144

[7] Lokanan, M., & Maddhesia, V. (2024). Supply chain fraud prediction with machine learning and artificial intelligence. *International Journal of Production Research*. https://doi.org/10.1080/00207543.2024.2361434

[8] Sebbar, A., & Zkik, K. (2023). Enhancing resilience against DDoS attacks in SDN-based supply chain networks using machine learning. *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, 230–234. https://doi.org/10.1109/CoDIT58514.2023.10284387

[9] Kim, J., et al. (2022). Securing third-party services in supply chains. *Supply Chain Security Journal*, ***14***(1), 112–123.

[10] Masum, M., Nazim, M., Faruk, M., Shahriar, H., Valero, M., Khan, M., Uddin, G., Barzanjeh, S., Saglamyurek, E., Rahman, A., & Ahamed, S. (2022). Quantum machine learning for software supply chain attacks: How far can we go? *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 530–538. https://doi.org/10.48550/arXiv.2204.02784

[11] Kumar, A., et al. (2021). Machine learning techniques for modern cybersecurity challenges. *Journal of Information Security Research*, ***12***(3), 45–67.

[12] Liu, H., Hu, B., & Zhang, J. (2021). Feature selection for machine learning: A survey. *Journal of Computational Intelligence*.

[13] Perumal, S., Sujatha, P., S., K., & Krishnan, M. (2024). Clusters in chaos: A deep unsupervised learning paradigm for network anomaly detection. *Journal of Network and Computer Applications*, ***235***, 104083. https://doi.org/10.1016/j.jnca.2024.104083

[14] Yeboah-Ofori, A., & Boachie, C. (2019). Malware attack predictive analytics in a cyber supply chain context using machine learning. *2019 International Conference on Cyber Security and Internet of Things (ICSIoT)*, 66–73. https://doi.org/10.1109/ICSIoT47925.2019.00019

[15] Sarker, I. H., et al. (2022). Role of machine learning in enhancing cybersecurity. *Cybersecurity and Privacy Journal*, ***5***(1), 12–34.

[16] Nguyen, T., et al. (2021). Anomaly detection in network traffic using unsupervised learning techniques. *Cybersecurity Review*, ***18***(2), 23–39.

[17] Jaber, A. (2024). Transforming cybersecurity dynamics: Enhanced self-play reinforcement learning in intrusion detection and prevention system. *2024 IEEE International Systems Conference (SysCon*), 1–8. https://doi.org/10.1109/SysCon61195.2024.10553626

[18] Zhao, Z., & Li, X. (2020). Machine learning approaches for cybersecurity: Applications and challenges. *Journal of Cybersecurity*, ***8***(4), 129–145. https://www.journals.elsevier.com/journal-of-cybersecurity

[19] Sharma, R., & Gupta, N. (2022). Adaptive authentication systems using machine learning. *Advances in Cybersecurity*, ***10***(4), 67–81.

[20] Lim, W., Yong, K., Lau, B., & Tan, C. (2024). Future of generative adversarial networks (GAN) for anomaly detection in network security: A review. *Computers & Security*, ***139***, 103733.

[21] Akter, S., Hossain Faruk, M. J., Anjum, N., Masum, M., Shahriar, H., Sakib, N., Rahman, A., Wu, F., & Cuzzocrea, A. (2022). Software supply chain vulnerabilities detection in source code: Performance comparison between traditional and quantum machine learning algorithms. *Proceedings of the 2022 IEEE International Conference on Big Data (Big Data*), 5639–5645. https://doi.org/10.1109/BigData55660.2022.10020813

[22] Al-Ansari, A., & Alsubait, T. (2022). Predicting cyber threats using machine learning for improving cyber supply chain. *2022 Fifth National Conference of Saudi Computers Colleges (NCCC*), 123–130. https://doi.org/10.1109/NCCC57165.2022.10067692

[23] Gokkaya, Z., et al. (2024). Software supply chain: Review of attacks, risk assessment strategies, and security controls. *International Journal of Safety and Security Engineering*, ***11***(5). https://doi.org/10.18280/ijsse.110505

[24] Cai, Z., Huang, S., Chen, C., Lin, J., & Ou, Y. (2023). Detecting supply chain attacks with unsupervised learning. *2023 9th International Conference on Applied System Innovation (ICASI*), 232–234. https://doi.org/10.1109/ICASI57738.2023.10179583

[25] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, ***3***, 1157–1182.

[26] Wloomis.(2023,September 27). Software Supply Chain Security: The Dataset.DFRLab. https://dfrlab.org/2023/09/27/software-supply-chain-security-the-dataset]

[27] Ghozi, W., Setiono, O., Rafrastara, F., Rijati, N., & Shidik, G. (2025). Tree-based ensemble algorithms and feature selection method for intelligent distributed denial of service attack detection. *Journal of Cyber Security and Mobility*, 14(1), 1–24. https://doi.org/10.13052/jcsm2245-1439.1411

[28] Korbaa, O., Jemili, F., & Meddeb, R. (2023). Intrusion detection based on ensemble learning for big data classification. *Cluster Computing*, 27(3), 3771–3798. https://doi.org/10.1007/s10586-023-04168-7

[29] Lim, W., Yong, K., Lau, B., & Tan, C. (2024). Future of generative adversarial networks (GAN) for anomaly detection in network security: A review. *Computers & Security*, 139, 103733. https://doi.org/10.1016/j.cose.2024.103733