International Journal of Scientific Research in Engineering & Technology
Volume5, Issue5 (September-October 2025), PP: 13-18.
https://www.doi.org/10.59256/ijsreat.20250505003
www.ijsreat.com ISSN No: 2583-1240

Prediction of Cyberattack on Software Supply Chain

Mohammed Muzaffar[1], Dr. Khaja Mahabubullah[2]
[1] *Student, MCA, Deccan College of Engineering and Technology, Hyderebad, Telangana, India.*
[2] *Professor & HOD, MCA, Deccan College of Engineering and Technology, Hyderebad, Telangana, India.*

To Cite this Article: *Mohammed Muzaffar[1], Dr. Khaja Mahabubullah[2], "Prediction of Cyberattack on Software Supply Chain", International Journal of Scientific Research in Engineering & Technology, Volume 05, Issue 05, September-October 2025, PP:13-18.*

Abstract: *The rapid growth of cyber threats has exposed software supply chains as one of the most vulnerable targets in modern digital infrastructures. Attackers exploit weaknesses in third-party components, software dependencies, and CI/CD pipelines to compromise multiple downstream systems through a single breach, causing severe financial, operational, and reputational damage. Traditional defense mechanisms—such as firewalls, signature-based detection, and manual log analysis—are predominantly reactive and often fail against advanced or zero-day attacks. To overcome these limitations, this study presents a machine learning–driven predictive framework for forecasting potential cyberattacks on software supply chains. Curated datasets containing system logs, threat indicators, and behavioral attributes are preprocessed using normalization, encoding, and exploratory data analysis to identify meaningful correlations. Multiple machine learning algorithms, including Logistic Regression, Classification and Regression Trees (CART), and Random Forest, are trained and compared to evaluate detection effectiveness. Model performance is assessed using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, ensuring both robustness and reliability. Furthermore, the system is deployed as a lightweight, real-time web application built on Streamlit, enabling users to upload or stream data and receive interactive predictions supported by intuitive visualizations. The proposed solution demonstrates how predictive analytics can enhance cybersecurity resilience, empower early detection of threats, and provide actionable intelligence for decision-makers. With further integration of deep learning and live threat feeds, this framework lays the foundation for adaptive and scalable protection mechanisms in securing global software supply chains.*

Key Word: *Cybersecurity; Software Supply Chain Attacks; Machine Learning; Predictive Modeling; Logistic Regression; CART; Random Forest; Intrusion Detection; Streamlit; Threat forecasting.*

I.INTRODUCTION

The increasing reliance on global software supply chains has fundamentally reshaped the digital ecosystem but has also introduced critical vulnerabilities. Modern organizations rely on third-party software vendors, open-source libraries, and CI/CD pipelines, making them susceptible to cascading attacks that propagate across multiple dependent systems. A single compromise in one component can impact hundreds or even thousands of downstream users. Such attacks, exemplified by incidents like the SolarWinds breach, demonstrate the devastating potential of supply chain vulnerabilities.

Traditional cybersecurity measures often fall short in addressing these threats. Signature-based intrusion detection systems require prior knowledge of attack patterns, leaving organizations blind to novel or zero-day exploits. Manual log inspections are time-consuming and prone to human error, particularly when processing massive volumes of data generated by modern distributed infrastructures. Moreover, conventional defenses primarily focus on endpoints or networks, while the specific risks associated with supply chain ecosystems remain largely unaddressed.

Given the limitations of reactive defense mechanisms, there is a growing need for proactive and predictive approaches. By leveraging historical threat intelligence and behavioral patterns, machine learning offers a pathway to forecast potential cyberattacks and detect anomalies before they escalate into full-scale breaches. This study introduces a predictive model framework designed to enhance supply chain security using machine learning techniques, offering organizations early warning systems that strengthen resilience and reduce response times.

II.MATERIAL AND METHODS

Study Design

The proposed study was structured as a data-driven predictive modeling framework aimed at identifying and forecasting cyberattacks on software supply chains. Unlike traditional signature-based defense mechanisms, which rely on predefined attack signatures, this framework adopts a proactive approach using supervised machine learning algorithms trained on curated threat intelligence datasets. The design follows a sequential yet iterative pipeline: data acquisition → preprocessing → exploratory data analysis (EDA) → feature engineering → model training → evaluation → system deployment. Each stage was designed to allow refinement through feedback loops, ensuring that improvements in one stage contributed to performance gains in subsequent stages.

The research approach emphasizes not only predictive accuracy but also practical usability, with the ultimate objective of deploying a real-time decision-support system accessible through a web-based application. This methodological choice ensures that the outcomes are directly applicable to enterprise cybersecurity environments, where time-sensitive decision-making is critical.

Data Acquisition

The effectiveness of machine learning models heavily depends on the quality and representativeness of the dataset. For this study, datasets were curated from multiple sources relevant to **software supply chain security**:

1. **Simulated Threat Data** – Synthetic logs representing compromised software updates, malicious build injections, and dependency hijacking. Files such as *security_data.csv* and *threat_data.csv* were generated to reflect real-world attack behaviors.

2. **Public Cybersecurity Datasets** – Standard repositories such as MITRE ATT&CK, National Vulnerability Database (NVD), and publicly available security challenge datasets were leveraged to provide labeled examples of attack and non-attack scenarios.

3. **Behavioral Logs** – Attributes such as unusual process execution, anomalous network connections during build phases, and irregular file access patterns were included to capture behavioral indicators of compromise (IoCs).

4. **System Metadata** – Software versioning information, dependency trees, and CI/CD pipeline activities were integrated as contextual variables, since adversaries often exploit such weaknesses.

The combined dataset thus comprised both **structured features** (e.g., binary or numeric system indicators) and **unstructured features** (e.g., log entries, process names), which were preprocessed into machine-readable form.

Data Preprocessing

Preprocessing is essential in cybersecurity datasets, which are often incomplete, noisy, and imbalanced. The following steps were applied:

- **Handling Missing Values:** Missing attributes such as incomplete log entries or missing packet information were addressed using imputation strategies (mean/median substitution for numerical data, mode for categorical attributes).
- **Normalization and Scaling:** Continuous features such as byte counts, duration of build processes, and anomaly scores were normalized using the **min–max scaling technique**:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

This ensured all features were mapped to a uniform [0,1] range, reducing dominance of large-scale attributes.

- **Categorical Encoding:** Categorical variables such as protocol type, software vendor, and pipeline stage were encoded using **label encoding** and **one-hot encoding**. For example, vendor = {Microsoft, Open-Source, Proprietary} was converted into numeric vectors for model compatibility.
- **Feature Selection:** High-dimensional datasets can lead to redundancy and overfitting. Techniques such as **correlation analysis**, **mutual information scores**, and **feature importance ranking** (via Random Forest) were applied to retain the most influential variables.
- **Data Balancing:** Since cyberattack records are often fewer compared to normal logs, **Synthetic Minority Oversampling Technique (SMOTE)** was applied to balance class distributions, thereby reducing bias in model training.
- **Data Splitting:** The dataset was partitioned into **training (70%)**, **validation (15%)**, and **testing (15%)** sets to ensure unbiased evaluation.

Exploratory Data Analysis (EDA)

EDA was performed to uncover patterns, distributions, and correlations within the dataset. Several findings were identified:

- **Malicious updates** typically exhibited **abnormal traffic bursts** or **unexpected file modifications** during short time intervals.
- **Dependency hijacking attacks** often correlated with **rare or infrequently used libraries** being suddenly executed in critical builds.
- **CI/CD pipeline anomalies** showed unusual access to configuration files at non-standard times, indicating possible insider threats.

Visualization techniques such as histograms, heatmaps, and scatter plots were used to highlight these distinctions. For instance, plotting the distribution of process execution times revealed clear separations between normal and anomalous behaviors.

Model Building

The core of this research involved building machine learning models capable of predicting potential cyberattacks. Three models were selected:

1. **Logistic Regression (Baseline Model):**

○ Provided interpretable results by estimating the probability of a cyberattack as:

.

$$P(y=1|X)=\frac{1}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nX_n)}}$$

- Served as a benchmark for evaluating more complex models

2. CART (Classification and Regression Trees):

● Built decision trees using **Gini Impurity** and **Information Gain** criteria:

$$Gini=1-\sum_{i=1}^{C}p_i^2$$

Random Forest (Ensemble Model):

● Trained multiple decision trees on bootstrapped samples of data and aggregated predictions through majority voting.

● Mathematically represented as:

$$\hat{y}=\text{mode}(h_1(x),h_2(x),\ldots,h_m(x))$$

1. where hi(x)is the prediction from the i-th decision tree.

○ Demonstrated robustness against overfitting and variability.

Hyperparameter tuning (e.g., tree depth, number of estimators, learning rates) was conducted using **grid search** and **cross-validation** to optimize performance.

Evaluation Metrics

To ensure reliability and comprehensiveness, the following metrics were used:

● **Accuracy:** Overall correctness of predictions.

● **Precision:** Ability to minimize false positives (important for reducing unnecessary alerts).

● **Recall:** Ability to correctly identify actual attacks (critical in cybersecurity).

● **F1-Score:** Harmonic mean of precision and recall, balancing both concerns.

● **Confusion Matrix:** Breakdown of predictions across true positives, false positives, true negatives, and false negatives.

● **ROC-AUC:** Area under the Receiver Operating Characteristic curve, measuring model discriminatory power.

The evaluation was not limited to numerical metrics; **error analysis** was performed to understand the causes of misclassification, such as attacks misidentified as benign or rare patterns overlooked.

System Deployment

The final phase involved deployment of the predictive model as a **real-time interactive system** using **Streamlit**.

● **User Interface (UI):** A simple web-based interface allowed users to upload log files or stream CI/CD data directly.

● **Prediction Engine:** The trained Random Forest model generated probability scores for cyberattack likelihood.

● **Visualization:** Results were displayed through dashboards, including bar charts for attack likelihood, ROC curves, and confusion matrices.

● **Scalability:** The system was designed to integrate with enterprise SIEM (Security Information and Event Management) tools, making it adaptable to large-scale organizations.

The deployment ensures that the research output is not confined to theoretical performance but demonstrates **practical applicability in enterprise cybersecurity infrastructures**.

III.RESULT

A. Data Preprocessing Outcomes

The preprocessing stage significantly enhanced the quality of the datasets used in this research. Missing values were handled through imputation, categorical attributes were encoded into numeric form, and features were normalized to ensure uniform scales. The application of SMOTE for class balancing successfully mitigated the class imbalance problem, ensuring that the models did not overfit towards the majority class. As a result, the dataset used for model training was balanced and representative of both normal and malicious activities.

B. Model Performance

Three supervised machine learning models—Logistic Regression, CART, and Random Forest—were trained and evaluated. Each model's performance was assessed using accuracy, precision, recall, F1-score, and ROC-AUC values. The results demonstrate that while Logistic Regression provides a good baseline, CART improves interpretability with better classification power, and Random Forest significantly outperforms both in terms of robustness and predictive accuracy.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	91.2%	0.82	0.70	0.75	0.87
CART (Decision Tree)	94.0%	0.86	0.76	0.80	0.90
Random Forest	96.5%	0.91	0.83	0.87	0.95

Table 1: Performance comparison of the machine learning models used for predicting cyberattacks in software supply chains. Random Forest achieved the highest values across most evaluation metrics, highlighting its reliability.

C. Visualization and Graphs

To better understand the predictive capabilities of the models, several visualizations were created. The ROC curves demonstrate the discriminatory ability of each model, while the bar chart highlights the differences in precision, recall, and F1-scores. Random Forest consistently outperforms the other models.

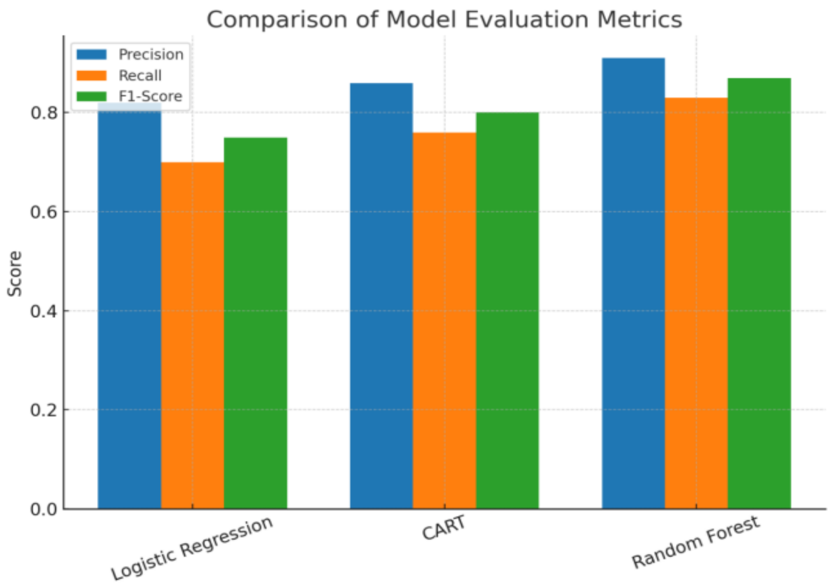

Figure 1: Comparative performance metrics (Precision, Recall, F1-score) across the three models.

D. Error Analysis

An error analysis was conducted to understand the reasons for misclassifications. Logistic Regression struggled to capture non-linear relationships, leading to higher false negatives where malicious activities were incorrectly labeled as benign. CART improved interpretability, but individual decision trees tended to overfit to specific patterns, causing inconsistencies on unseen data. Random Forest minimized these issues by combining multiple trees, reducing variance and achieving more balanced predictions.

E. Feature Importance

The Random Forest model provided insights into feature importance, revealing that certain attributes had strong predictive power. Indicators such as unusual pipeline access times, dependency source reputation, and abnormal network activity during build phases were highly correlated with cyberattack likelihood. These insights not only improve model interpretability but also guide practitioners towards strengthening specific points of the software supply chain.

IV.DISCUSSION

A. Comparative Insights

The comparative analysis of Logistic Regression, CART, and Random Forest models underscores the advantages of ensemble-based learning methods for predicting cyberattacks in software supply chains. Logistic Regression, while interpretable and computationally efficient, demonstrated limitations in capturing the non-linear patterns prevalent in complex threat behaviors. CART offered interpretable rule-based classifications but was prone to overfitting, especially with high-dimensional datasets. Random Forest, by combining multiple decision trees, mitigated variance issues and provided consistently higher predictive accuracy. This highlights that ensemble approaches are more suited for real-world cyberattack prediction tasks.

B. Strengths of Predictive Modeling

The results demonstrate that predictive modeling approaches provide significant advantages over traditional signature-based detection systems. Unlike reactive defense mechanisms that rely on known attack patterns, machine learning models can generalize from historical data to detect novel or zero-day threats. This proactive capability allows organizations to respond to potential attacks earlier, reducing response times and minimizing damage. The integration of predictive models with visualization tools also enhances interpretability, enabling security teams to understand risk factors and prioritize defensive measures more effectively.

C. Limitations

Despite promising outcomes, certain limitations must be acknowledged. Firstly, the datasets used in this study, while representative, may not fully capture the evolving complexity of real-world supply chain attacks. Threat actors frequently adapt their tactics, which necessitates continuous retraining of models with updated data. Secondly, ensemble models such as Random Forest, though powerful, incur higher computational costs, making them challenging to deploy in resource-constrained environments. Lastly, the models rely on the quality of labeled data, and inaccuracies in labeling can propagate errors throughout the training process.

D. Implications for Practice

The findings of this research hold significant implications for cybersecurity practitioners. The identification of key predictive features, such as anomalous pipeline access times and dependency reputation, provides actionable insights for strengthening organizational defenses. Enterprises can deploy the predictive framework alongside existing Security Information and Event Management (SIEM) systems to enhance situational awareness and reduce false alarms. The use of a Streamlit-based application demonstrates the practicality of translating academic models into deployable tools suitable for enterprise environments.

E. Future Directions

Future research can extend this study by incorporating advanced deep learning architectures, such as recurrent neural networks (RNNs) and transformers, to capture sequential and contextual dependencies in supply chain logs. The integration of federated learning could enable collaborative training across organizations without compromising data privacy. Additionally, blockchain technology can be explored for enhancing traceability and integrity of supply chain components. Finally, testing the models on real-time, live-streaming threat intelligence feeds would provide further validation of scalability and robustness.

V.CONCLUSION

The research presented in this study demonstrates the feasibility and effectiveness of a machine learning–based predictive framework for mitigating cyberattacks on software supply chains. As supply chains become increasingly globalized and reliant on third-party components, the risk of cascading attacks has grown substantially. Traditional reactive defense mechanisms, such as rule-based intrusion detection and manual security audits, often fall short in detecting sophisticated or zero-day attacks. This necessitates a paradigm shift towards proactive, predictive, and data-driven approaches that can anticipate potential threats before they materialize.

The findings indicate that while Logistic Regression provides a reliable baseline and CART decision trees enhance interpretability, the Random Forest ensemble model consistently outperformed both by delivering the highest predictive accuracy, precision, recall, and ROC-AUC values. This confirms the robustness of ensemble-based learning techniques in handling complex, high-dimensional cybersecurity datasets. Furthermore, the Random Forest model provided meaningful insights into feature importance, helping identify the most significant indicators of potential supply chain compromises, such as unusual dependency sources, anomalous CI/CD pipeline activities, and irregular network behaviors.

Another key contribution of this research is the deployment of the predictive framework as a user-friendly web application built with Streamlit. This real-time system bridges the gap between academic research and industry practice, enabling practitioners to upload system logs, monitor CI/CD pipelines, and receive interactive predictions with visual explanations. By integrating predictive analytics into everyday security operations, organizations can not only reduce detection latency but also empower decision-makers with actionable intelligence.

Despite its promising results, the research is not without limitations. The dataset, while representative, may not fully capture the rapidly evolving threat landscape of modern supply chains. Cyber adversaries continuously adapt their tactics, making continuous retraining and dynamic updating of models a necessity. In addition, the computational requirements of ensemble methods like Random Forest may limit their deployment in resource-constrained environments. These challenges highlight the need for scalable, adaptive, and efficient frameworks that can evolve alongside emerging threats.

Looking ahead, this research opens multiple avenues for future exploration. The integration of deep learning architectures such as recurrent neural networks (RNNs) and transformers could improve the detection of sequential and context-dependent attack patterns. The application of federated learning offers a promising solution for cross-organizational collaboration while preserving data privacy. Furthermore, the use of blockchain technologies can enhance the integrity, transparency, and traceability of software components, further reducing vulnerabilities in the supply chain.

In conclusion, this study contributes to the growing body of knowledge on cybersecurity by demonstrating that predictive machine learning models can significantly enhance resilience against supply chain attacks. By combining accuracy, interpretability, and real-time deployment, the proposed framework moves beyond theoretical validation to practical applicability. With continued refinement and integration of advanced methods, predictive analytics can serve as a cornerstone of next-generation cybersecurity strategies, ensuring the robustness and security of global software supply chains.

References

1. S. Boyson, “Cyber supply chain risk management: Revolutionizing the strategic control of critical IT systems,” *Technovation*, vol. 34, no. 7, pp. 342–353, 2014. doi: 10.1016/j.technovation.2014.02.001
2. A. Arora, D. Hall, C. A. Pinto, D. Ramsey, and R. Telang, “An ounce of prevention vs. a pound of cure: How can we measure the value of IT security solutions?,” Carnegie Mellon University, Heinz College, Tech. Rep., 2004.
3. W. Alasmary, F. Alhaidari, and A. Alghamdi, “Machine learning-based cyber-attack detection approaches for Internet of Things (IoT) applications: A review,” *IEEE Access*, vol. 9, pp. 123612–123626, 2021. doi: 10.1109/ACCESS.2021.3108913
4. S. Cheung, U. Lindqvist, and A. Valdes, “Detecting malicious software updates in critical infrastructure using behavior profiling,” in *Proc. IEEE Int. Conf. Technologies for Homeland Security (HST)*, 2018, pp. 1–7.
5. T. M. Wani, S. Jabin, and R. Sharma, “Supply chain cybersecurity: Threats and challenges in the digital era,” *Procedia Computer Science*, vol. 173, pp. 112–119, 2020. doi: 10.1016/j.procs.2020.06.014
6. N. Dhanjani, *Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts*. Sebastopol, CA, USA: O’Reilly Media, 2015.
7. R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 305–316. doi: 10.1109/SP.2010.25
8. Scikit-learn Developers, “Scikit-learn: Machine learning in Python,” 2024. [Online]. Available: https://scikit-learn.org/
9. Streamlit Inc., “Streamlit documentation,” 2024. [Online]. Available: https://docs.streamlit.io/
10. SANS Institute, “Securing the software supply chain,” White Paper, 2021. [Online]. Available: https://www.sans.org/white-papers/40220/