



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Cyber Threat Detection Using Machine Learning

S. Ramalakshmi, E Julie Ruth

Department of Computer Applications and Research Centre, Sarah Tucker College (Autonomous), Tirunelveli-7.

ABSTRACT

Cyber threats and unauthorized login attempts are major security concerns in modern web applications. Traditional username and password authentication may not provide sufficient protection when valid credentials are used from unfamiliar devices or networks. This project presents a Cyber Threat Detection Using Machine Learning system that combines machine learning model evaluation, IP address verification, OTP authentication, email alerts, and SQLite database logging. The dataset contains login-related attributes such as device type, login method, authentication type, failed attempts, multi-factor authentication status, suspicious activity, and IP address information. The collected data is preprocessed and used to train and compare four classification algorithms: Random Forest, Support Vector Machine (SVM), Logistic Regression, and K-Nearest Neighbors (KNN). The recorded experimental results show that Random Forest achieved the highest accuracy of 99.84%, followed by KNN at 99.52%, Logistic Regression at 99.21%, and SVM at 83.15%. Therefore, Random Forest was selected as the best-performing model based on the recorded accuracy results. The web application is developed using Python and Flask. It validates login credentials and checks the user's IP address to determine whether additional authentication is required. Authorized login requests receive direct dashboard access, while login requests from unauthorized IP addresses activate a four-digit OTP verification process. The OTP and a security alert are sent to the registered email address. Incorrect OTP entries display a separate error page with a Try Again option. Login activities are recorded in an SQLite database for monitoring and analysis. The proposed system demonstrates a practical approach to strengthening login security through machine learning evaluation and layered authentication.

Keywords— Cybersecurity, Machine Learning, Threat Detection, IP Verification, OTP Authentication.

1. INTRODUCTION

The rapid growth of internet technologies, online transactions, and web-based applications has increased the importance of cybersecurity. Unauthorized access may occur when attackers obtain valid login credentials or attempt to access accounts through unfamiliar devices and networks. Traditional authentication systems mainly depend on usernames and passwords, which may not be sufficient to identify suspicious login activities. Machine learning provides techniques for analysing login-related data and classifying records based on learned patterns. The proposed project focuses on cyber threat detection by considering security features such as device type, login method, authentication type, failed attempts, multi-factor authentication status, suspicious activity, and IP address information.

To improve login security, the proposed system combines machine learning model evaluation with IP address verification, OTP authentication, email notifications, and database logging. Four machine learning algorithms are compared to identify the best-performing model based on the recorded experimental results. The web application is developed using Python and Flask, while SQLite stores login and security-related records. When valid credentials are entered from an authorized IP address, the system allows direct dashboard access. If the login request originates from an unauthorized IP address, the system requires additional OTP verification and sends a security alert by email. This layered approach supports secure login processing and helps monitor potentially unauthorized access attempts.

2. LITERATURE REVIEW

Buczak and Guven surveyed data mining and machine learning methods used for cybersecurity intrusion detection. Their work highlights the importance of intelligent techniques for analysing security-related data and identifying potential threats [1]. Sommer and Paxson discussed the challenges of applying machine learning to real-world network intrusion detection, including the need for realistic evaluation and careful interpretation of results [2].

Tavallae et al. analysed the KDD Cup 99 dataset and demonstrated the importance of understanding dataset characteristics when developing intrusion detection systems [3]. Khraisat et al. reviewed intrusion detection techniques, datasets, and challenges, providing an overview of the different approaches used in security monitoring [4]. Sharafaldin et al. discussed intrusion detection dataset development and traffic characterization, emphasizing the importance of representative data for evaluating detection methods [5].

These studies motivated the selection of a machine-learning-based cybersecurity project. The present work applies machine learning concepts to login-security data and combines model comparison with IP verification, OTP authentication, email alerts, and database logging in a web application.

3. RESEARCH GAP AND OBJECTIVES

3.1 Research Gap

Traditional login systems primarily depend on username and password verification. Such systems may require additional checks when valid credentials are used from unfamiliar IP addresses or devices. Fixed security rules may also have limitations when login behaviour changes. Therefore, a practical system is needed that combines login-related feature analysis, machine learning model comparison, IP verification, additional authentication, and security activity logging.

The present project addresses this need by evaluating machine learning algorithms on login-related data and combining the security application with IP verification, OTP authentication, email alerts, and SQLite logging.

3.2 Objectives

The main objectives of this project are:

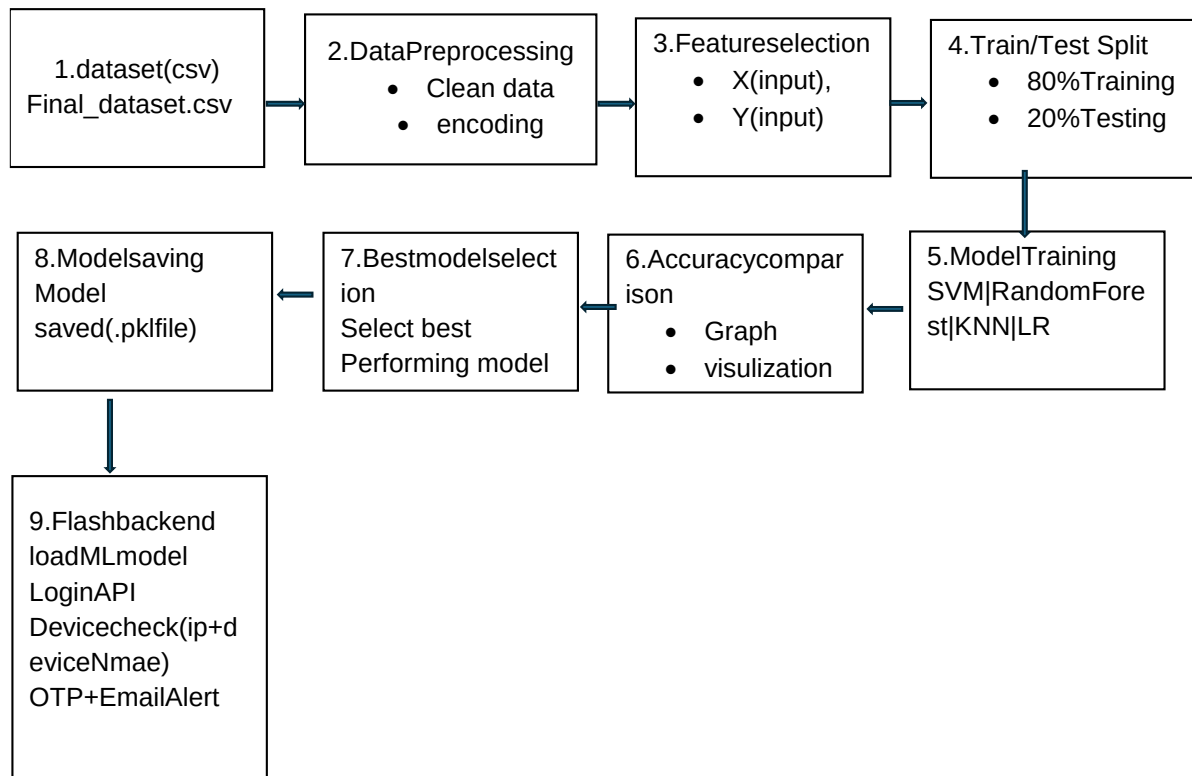
1. To develop a cyber threat detection system using machine learning.
2. To prepare and analyse login-related security data.
3. To train and compare four machine learning classification algorithms.
4. To identify the best-performing algorithm based on experimental results.
5. To verify login credentials and check IP address authorization.
6. To provide OTP verification for login requests from unauthorized IP addresses.
7. To send OTPs and security alerts through email.
8. To store login and security-related information in an SQLite database.

4. PROPOSED METHODOLOGY

The proposed methodology begins with collecting and preparing login-related security data. The dataset is cleaned and transformed into a suitable format for machine learning. Relevant features, including device type, login method, authentication type, failed attempts, MFA status, suspicious activity, and IP address components, are selected for model development.

The prepared dataset is used to train and evaluate Random Forest, Support Vector Machine, Logistic Regression, and K-Nearest Neighbors. Their accuracy values are compared to identify the best-performing model. Based on the recorded results, Random Forest achieved the highest accuracy of 99.84% and was saved as `best_model.pkl`.

The web application is developed using Python and Flask. During login, the system validates the username and password and checks the IP address. Authorized IP addresses allow direct dashboard access. Unauthorized IP addresses activate the OTP verification process. A four-digit OTP is sent to the registered email with a security alert. The OTP is stored temporarily in the Flask session, and login activities are recorded in SQLite.

System Workflow**5. SYSTEM DESCRIPTION**

The Cyber Threat Detection Using Machine Learning project is a web-based security application designed to support the identification and handling of suspicious login attempts. It combines machine learning model development and evaluation with credential verification, IP address checking, OTP authentication, email notification, and SQLite database logging.

The project includes dataset preparation, in which login and security-related records are cleaned and prepared for model training. Four machine learning classification algorithms are trained and compared using their recorded performance results. The application validates the username and password entered by the user and checks the login source against the authorized IP configuration. Login requests from unauthorized IP addresses require additional OTP verification, and the OTP and security alert are sent to the registered email address. Login and security-related information is stored in SQLite, and the user proceeds to the dashboard after completing the required authentication checks.

The system provides direct dashboard access for valid login requests from authorized IP addresses. If valid credentials are used from an unauthorized IP address, the system activates OTP verification. A correct OTP permits the user to proceed, while an incorrect OTP displays a Wrong OTP page with a Try Again option.

6. EVALUATION METRICS**6.1 Random Forest**

Random Forest is an ensemble machine learning algorithm that combines the predictions of multiple decision trees to produce a final classification. It can analyse several input features and is commonly used for classification problems. In this project, Random Forest was evaluated using the prepared login-security dataset.

The recorded experiment showed that Random Forest achieved an accuracy of 99.84%. It obtained the highest accuracy among the four tested algorithms and was selected as the best-performing model. The trained model was saved as best_model.pkl.

6.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm that identifies a decision boundary for separating different classes. It can be used to classify login records based on their security-related features.

In the recorded experiment, SVM achieved an accuracy of 83.15%. This was lower than the accuracy achieved by the other three algorithms in the comparison. The result indicates that SVM performed less accurately than the other tested models on the evaluated dataset.

6.3 Logistic Regression

Logistic Regression is a supervised classification algorithm that estimates the probability of a record belonging to a particular class. It is commonly used for binary classification tasks, such as distinguishing between normal and suspicious login records.

In this project, Logistic Regression achieved an accuracy of 99.21%. Its recorded accuracy was high, but it was lower than the accuracy of Random Forest and KNN in the comparison.

6.4 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that classifies a record by considering the classes of its nearest neighbouring records. The classification depends on the similarity between the input record and the neighbouring examples in the dataset.

In the recorded experiment, KNN achieved an accuracy of 99.52%. It performed better than Logistic Regression and SVM in the comparison, but its accuracy was slightly lower than Random Forest.

6.5 Evaluation Measures

Machine learning classification models must be evaluated to understand how effectively they classify records. Accuracy is used to compare the four algorithms in this project. Precision, recall, F1-score, and the confusion matrix are additional metrics that can be used to evaluate classification performance when their corresponding results are available

Accuracy: Accuracy measures the proportion of correctly classified records among all tested records.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Precision measures the proportion of records predicted as suspicious that are actually suspicious.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Recall measures the proportion of actual suspicious records correctly identified by the model.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: The F1-score combines precision and recall into a single measure.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Confusion Matrix: A confusion matrix summarizes classification results using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Machine Learning Model	Accuracy (%)
Random Forest	99.84%
K-Nearest Neighbors (KNN)	99.52%
Logistic Regression	99.21%
Support Vector Machine	83.15%

7. SYSTEM IMPLEMENTATION

The system implementation combines dataset preparation, machine learning model training, Flask-based login authentication, IP verification, OTP verification, email alerts, and SQLite database logging.

7.1 Dataset Collection and Preparation

The dataset contains login and security-related attributes such as device type, login method, authentication type, successful login status, failed attempts, MFA status, suspicious activity, and IP information. The IP address is represented using four components, IP1, IP2, IP3, and IP4, for processing.

7.2 Data Preprocessing and Feature Selection

The collected data is cleaned and transformed into a suitable format for machine learning. Relevant attributes are selected to represent login behaviour and security information. The prepared data is then used for model training and evaluation.

7.3 Model Training and Selection

Four classification algorithms—Random Forest, KNN, Logistic Regression, and SVM—are trained and compared. The recorded accuracy results are used to identify the best-performing algorithm. Random Forest achieved the highest accuracy of 99.84%, and the trained model was saved as `best_model.pkl`.

7.4 Login Authentication and IP Verification

The Flask application receives the username and password from the login page and validates the credentials. After successful credential verification, the system checks the user's IP address against the authorized IP configuration. A valid login from an authorized IP address is allowed direct dashboard access. A valid login from an unauthorized IP address activates additional authentication

7.5 OTP Verification and Email Alert

For an unauthorized IP address, the application generates a four-digit OTP and sends it to the registered email address with a security alert. The OTP is stored temporarily in the Flask session. The user must enter the OTP on the verification page. A correct OTP allows the user to proceed, while an incorrect OTP displays the Wrong OTP page with the message "Please enter the correct OTP" and a Try Again option.

8. RESULTS AND DISCUSSIONS

The proposed Cyber Threat Detection Using Machine Learning system was evaluated using login-related features with four machine learning algorithms: Random Forest, K-Nearest Neighbors (KNN), Logistic Regression, and Support Vector Machine (SVM). The algorithms were compared using their recorded accuracy values to identify the best-performing model for cyber threat classification.

The recorded experimental results show that Random Forest achieved the highest accuracy of **99.84%**, followed by K-Nearest Neighbors at **99.52%**, Logistic Regression at **99.21%**, and Support Vector Machine at **83.15%**. Based on these recorded accuracy values, Random Forest was selected as the best-performing model for the proposed system.

The dataset initially contained 5,000 login-related records. After data cleaning and preprocessing, 3,145 records were retained for further processing, while 1,855 records were removed during cleaning. The retained dataset included features such as device type, login method, authentication type, failed attempts, multi-factor authentication status, suspicious activity, and IP-related information. These features were prepared for machine learning model development and evaluation.

The comparison of the four machine learning algorithms is presented in a table containing their recorded accuracy values. A grouped bar chart can also be used to visually compare the accuracy of Random Forest, K-Nearest Neighbors, Logistic Regression, and Support Vector Machine. The comparison indicates that Random Forest achieved the highest recorded accuracy among the evaluated algorithms. Precision, recall, F1-score, and ROC-AUC comparisons can be included when the corresponding experimental results are available.

The selected model is saved as `best_model.pkl`. The web application is developed using Python and Flask. When a user enters login credentials, the application validates the credentials and checks the IP address against the authorized IP configuration. If valid credentials are entered from an authorized IP address, the application allows direct dashboard access. If valid credentials are entered from an unauthorized IP address, the system activates a four-digit OTP verification process and sends an email security alert. An incorrect OTP displays a Wrong OTP page with a Try Again option. Relevant login and security information is stored in an SQLite database for monitoring and analysis.

The results demonstrate the integration of machine learning model evaluation with additional login security mechanisms. The recorded comparison identifies Random Forest as the highest-accuracy model, while IP address verification, OTP authentication, email alerts, and database logging provide additional security and monitoring features in the implemented prototype. The recorded accuracy values should be confirmed using the final model evaluation output before submission.

9. CONCLUSION

The Cyber Threat Detection Using Machine Learning project presents a web-based approach to strengthening login security. It combines machine learning model evaluation with IP address verification, OTP authentication, email alerts, and SQLite database logging. Four machine learning algorithms were compared using the recorded accuracy results. Random Forest achieved the highest accuracy of 99.84%, followed by KNN, Logistic Regression, and SVM.

The Flask application validates login credentials and checks the login source. Authorized IP addresses receive direct dashboard access, while unauthorized IP addresses require additional OTP verification. Email notifications inform the user about the login attempt, and SQLite maintains login-related records for monitoring.

The project demonstrates how machine learning evaluation and layered authentication can support the development of a login-security application. Further testing with diverse datasets and additional performance measures can help evaluate its effectiveness more comprehensively.

10. FUTURE WORK

The system can be improved by collecting a larger and more diverse dataset containing different login patterns and suspicious activities. Additional machine learning algorithms and advanced classification techniques can be evaluated to determine whether they improve detection performance.

Future development can include real-time machine learning prediction within the login workflow, along with stronger IP reputation checks, device identification, unusual login-time analysis, and monitoring of repeated failed attempts. These features could provide more information when assessing suspicious login requests.

The application can also be enhanced with account lockout, improved session management, stronger OTP expiry and retry controls, and additional multi-factor authentication methods. An administrator dashboard could display login history, suspicious activity, IP information, and security alerts.

Finally, cloud deployment and periodic model retraining could be explored to support wider access and adapt the system to new login patterns. These enhancements may help develop a more comprehensive and adaptive login-security solution.

REFERENCES

- [1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.
- [2] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.
- [3] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
- [4] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, art. no. 20, 2019, doi: 10.1186/s42400-019-0038-7.
- [5] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116.