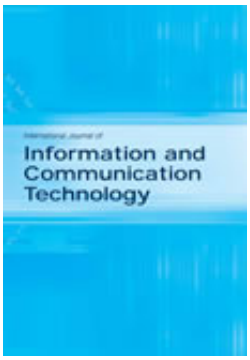
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

A machine learning based cyber security threat prediction algorithm for tourism hotels

Dan Fan

DOI: 10.1504/IJICT.2025.10070773

Article History:

Received:	11 March 2025
Last revised:	23 March 2025
Accepted:	23 March 2025
Published online:	08 May 2025

A machine learning based cyber security threat prediction algorithm for tourism hotels

Dan Fan

Dongguan Polytechnic,
Dongguan 2009067, China
Email: 2009067@dgpt.edu.cn

Abstract: In the travel and hospitality sector, digitalisation has brought network security threats. Particularly with big data and high-dimensional characteristics, traditional network security techniques find it difficult to control dynamic and complicated security threats. Popular research subjects based on performance are machine learning-based threat prediction methods and integrated learning approaches. This paper presents XG-CatSec, a machine learning (XGBoost and Catboost fusion) model to increase tourist and hospitality cybersecurity threat prediction accuracy and robustness. While CatBoost simplifies data preparation and optimises category feature processing, XGBoost increases model accuracy utilising gradient boosting trees. Combining these technologies in XG-CatSec raises the threat identification for hotel cybersecurity. XG-CatSec beats SVM and random forest in accuracy, precision, and recall on the NSL-KDD data. This report motivates further research by implying a special cybersecurity threat prediction solution for tourism and hospitality.

Keywords: XGBoost; CatBoost; tourist hotels; network security.

Reference to this paper should be made as follows: Fan, D. (2025) 'A machine learning based cyber security threat prediction algorithm for tourism hotels', *Int. J. Information and Communication Technology*, Vol. 26, No. 12, pp.89–103.

Biographical notes: Dan Fan received his Bachelor's in Central China Normal University of Tourism management in 2009. He is currently working in the Dongguan Polytechnic. His research interests include machine learning and tourism management.

1 Introduction

Cybersecurity has grown to be a major global issue given the fast expansion of information technology and the popularity of the Internet (Lu and Da, 2018; Kimani et al., 2019). Particularly in the travel and hotel sectors, where a lot of personal and financial data is kept and shared online, cybersecurity issues are growing and seriously endanger the hotel sector. Apart from handling conventional forms of cyberattacks such viruses and Trojans, the hotel sector also has to cope with more advanced cyber security concerns including distributed denial of service attacks (DDoS), SQL injections, and cross-site scripting attacks (XSS) (Chughtai et al., 2024).

Mazhar et al. (2023) have proposed many machine-learning-based approaches for cybersecurity threat identification if we are to properly avoid and react to these security concerns. These techniques can find possible security concerns and automatically extract characteristics from enormous volumes of data. But conventional cybersecurity detection techniques sometimes rely on rule matching and signature detection, which lack adaptability and the capacity to react to fresh threats. Because machine learning-based threat detection systems can automatically discover intricate patterns in data, they have progressively become a research focus in recent years (Sarker, 2023; Amiri et al., 2024).

There are still certain flaws even if current studies have been quite successful in several aspects. Decision trees and support vector machines (SVMs) are two classic machine learning methods that may efficiently classify and forecast; nevertheless, their performance is typically restricted for complicated cyber-attack patterns, particularly hidden attacks (Kandasamy and Roseline, 2025).

Particularly in the field of machine learning, academics have suggested several approaches to handle various kinds of assaults in recent years as the number of cybersecurity concerns rises. Usually based on rule matching and signature detection, which are more suited in handling known assaults, traditional cybersecurity threat detection techniques typically fail when confronted with new or unidentified threats. Azam et al. (2023) looked to machine learning-based threat detection techniques in order to address this problem. These techniques lack flexibility to new assaults and are inadequate in managing big-scale data and high-dimensional characteristics even if they can identify threats to some degree (Jha et al., 2024).

Integrated learning approaches have been extensively applied in cyber security threat identification recently. By aggregating several weak classifiers into strong ones, integrated learning approaches as random forest and Adaboost enhance the prediction performance of models. Integrated approaches may better manage data imbalance issues and complicated assault patterns as compared with a single model; they also offer more resilience and accuracy. Thanks in great part to their outstanding performance-especially when considering large-scale data and high-dimensional feature problems-advanced integrated learning algorithms as XGBoost and Catboost have been extensively applied in recent years in many classification challenges. While Catboost enhances the decision tree algorithm, especially when dealing with categorical features, greatly lowers the complexity of data preprocessing, which makes it perform even better on cybersecurity datasets, XGBoost shows outstanding ability in handling datasets with missing values and category imbalance.

Deep learning techniques have meanwhile progressively taken the stage in the world of cybersecurity as a research focus. Deep learning methods including deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) are extensively utilised for the identification and prediction of cybersecurity risks, and can automatically extract characteristics and categorise them from complicated data (Sarker, 2021; Abdi et al., 2024). Although deep learning approaches perform well in several cyber-attack detection tasks, their dependency on large-scale labelled data and extended training time, plus the poor interpretability of the models themselves, make deep learning methods still confront significant difficulties in practical implementations. To leverage their individual capabilities, several researchers have so sought to mix conventional machine learning with deep learning approaches. For some cybersecurity threat detection activities, for instance, hybrid models combining XGBoost and DNN have been demonstrated to dramatically increase performance; nonetheless, how to

develop more effective and simple-to-deploy hybrid models is still a hot issue (Mohan and Subathra, 2023).

The XG-CatSec model suggested in this work reflects mostly the following important innovations:

1 Fusion of XGBoost and CatBoost: In order to leverage their individual strengths and thereby increase the prediction performance of the model, this work presents in this work an original fusion technique combining two integrated learning algorithms, XGBoost and Catboost. While Catboost is able to efficiently lower the bias of the class features, XGBoost performs well in managing huge-scale data. Combining the two results in a model suggested in this work with better accuracy and resilience in cyber security threat prediction.

2 Security threat prediction model for tourism and hotel industry: In this work, it consider the features of the hotel and tourist sectors and apply particular modelling and prediction of cyber security risks. Important for enhancing the security of this sector is a customised network security threat detection approach suggested by considering the special network traffic and security threat patterns of the hotel sector.

3 Model optimisation strategy combining feature ablation and hyperparameter optimisation: This work combines hyperparameter optimisation with feature ablation in the tests to increase the model performance even further. The XG-CatSec model suggested in this study not only improves the prediction accuracy but also the interpretability and practicality of the model by progressively eliminating useless aspects and optimising the hyperparameters of the model.

2 Relevant technologies

2.1 XGBoost

XGBoost is a fast machine learning method derived from gradient boosting tree (GBT) optimised implementation (Dong et al., 2024; Dele-Afolabi et al., 2024). XGBoost builds a sequence of decision trees step-by-step, where the production of each tree depends on the error of the preceding tree and the residuals are continually corrected by an additive model, hence improving the prediction ability (see Figure 1).

XGBoost aims to minimise an objective function with two components: one portion is the training error (i.e., the loss function), and the other is a regularisation term preventing overfitting of the model (Bukowski et al., 2024; Shaik et al., 2024).

Constructing a sequence of trees $f_t(x)$ helps to minimise the objective function assuming n samples, where x_i are the characteristics of the i^{th} sample and y_i are the true labels. The last model follows:

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i) \tag{1}$$

XGBoost aims to reduce the objective function below:

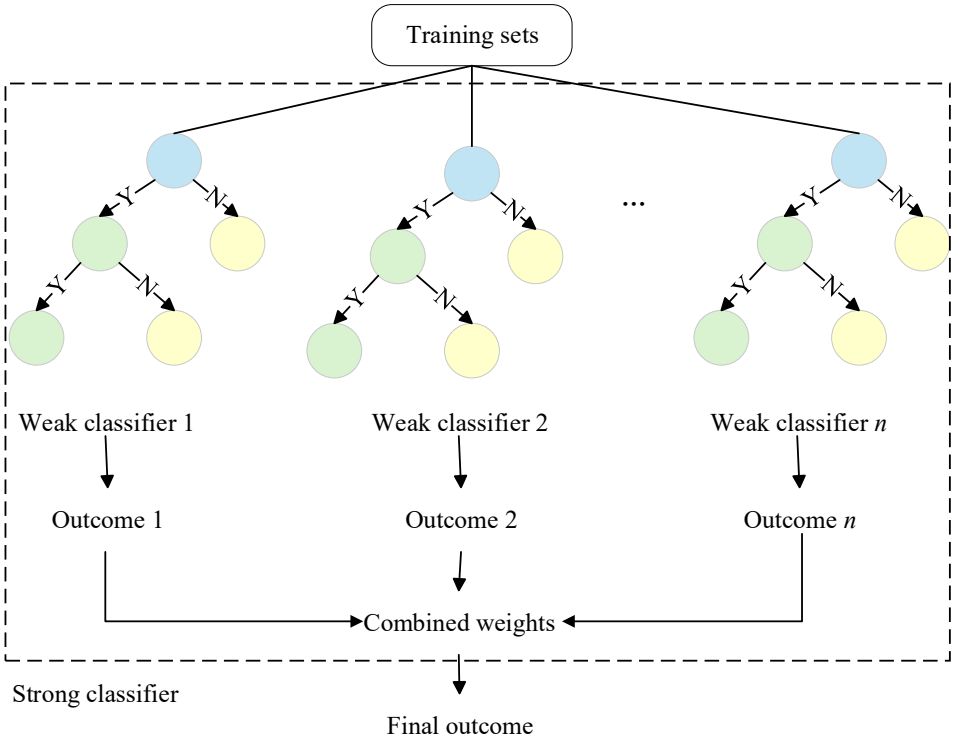
$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega(f) \tag{2}$$

where $l(y_i, \hat{y}_i)$ is the loss function – which gauges the variation between the actual label and the projected value – commonly used loss functions are either logarithmic or squared error. The regularisation term $\Omega(f)$ is used to prevent overfitting and regulate model complexity. Usually a complexity function of the tree, the regularisation term is represented as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{k=1}^{T} w_k^2 \tag{3}$$

where γ and λ are regularisation parameters, T is the tree's leaf node count, w_k is the weight of the k^{th} leaf node. XGBoost can properly manage tree complexity and prevent overfitting by using regularising words (Asselman et al., 2023).

Figure 1 XGBoost structure (see online version for colours)



XGBoost's training method is to step-by-step maximise the goal function (Li et al., 2022). Assuming that $t-1$ rounds of training have already been completed, the present model's prediction value is $\hat{y}_i^{(t-1)}$ XGBoost lowers the prediction error by including a fresh tree $f_t(x)$ with the updated prediction value:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{4}$$

XGBoest aims to reduce the following objective function in every training round:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)}\right) + f_t\left(x_i\right) + \Omega(f) \tag{5}$$

XGBoost uses Taylor expansion to do a second-order approximation of the loss function therefore optimising the objective function effectively. The objective function's approximate form follows from unfolding the loss function:

$$L^{(t)} \approx \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t\left(x_i\right)^2\right] + \Omega\left(f_t\right) \tag{6}$$

$$g_i = \frac{\partial l\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial \hat{y}_i} \tag{7}$$

$$h_i = \frac{\partial^2 l\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial \hat{y}_i^2} \tag{8}$$

where g_i is the loss function's first order gradient and h_i is its second order derivative. XGBoost can rapidly update the tree's leaf nodes using this gradient information.

XGBoost builds every tree in the training process by choosing the best split point (Demir and Sahin, 2023). XGBoost estimates the gain of every split point to choose the best one. The gain g may be stated assuming that the dataset D is split into two subsets D_L and D_R as:

$$g = \frac{1}{2}\left[\frac{\left(\sum_{i\in D_L} g_i\right)^2}{\sum_{i\in D_L} h_i + \lambda} + \frac{\left(\sum_{i\in D_R} g_i\right)^2}{\sum_{i\in D_R} h_i + \lambda} + \frac{\left(\sum_{i\in D} g_i\right)^2}{\sum_{i\in D} h_i + \lambda}\right] - \gamma \tag{9}$$

where γ and λ are regularisation parameters, g_i and h_i are the first- and second-order gradients of sample i in the left and right subsets respectively. XGBoost finds the best splitting point by increasing the gain, hence enhancing the tree's prediction capacity.

XGBoost further enhances the generalisation of the model by means of row and column sampling. Row sampling is the random selection of the training data; column sampling is the random selection of part of the characteristics for each tree throughout its development. These techniques help XGBoost lower overfitting and improve model resilience.

Furthermore maximised is XGBoost's computational efficiency. Especially with large-scale data, XGBoost may greatly shorten the training time by parallelising the calculation and speed the tree development process. Computational efficiency may be raised in every training cycle by assigning computing chores to several threads or computers.

XGBoost can avoid overfitting with the previous methods while guaranteeing excellent accuracy and effective processing of big-scale datasets. Its ability to swiftly create extremely accurate models and perform effectively in several tasks gives it benefits. XGBoost's characteristics help it to be a quite popular tool in data analytics and machine learning.

2.2 CatBoost

Based on gradient boosting decision trees (GBDT), CatBoost is a machine learning method especially designed for handling of category features (Zhang and Jánošík, 2024). CatBoost is able to effectively process category data, minimise information leakage, and limit the danger of overfitting by means of its goal coding and sorting algorithms, while conventional gradient boosting methods typically need complicated coding of category features while processing them (Chen et al., 2024).

Using the target coding approach, CatBoost transforms category characteristics into the average value of the target variable. Assume x_i is a category feature; category k values $x_i^{(k)}$; the target coding formula is:

$$T\left(x_i^{(k)}\right) = \frac{\sum_{j=1}^{m} y_j \cdot I\left(x_i^{(j)} = x_i^{(k)}\right)}{\sum_{j=1}^{m} I\left(x_i^{(j)} = x_i^{(k)}\right)} \tag{10}$$

where $I(x_i^{(j)} = x_i^{(k)})$ is the indicator function-1 when the category feature $x_i^{(k)}$ of sample j. matches the category value and 0 otherwise-y_j is the goal value of sample j.

CatBoost adds a smoothing mechanism to smooth the coded values of the less frequent category features to the global target mean T_{global}, therefore avoiding the overfitting of some values in the category features. The smoothing equation is:

$$T_{smooth}\left(x_i^{(k)}\right) = \frac{n_k \cdot T\left(x_i^{(k)}\right) + \lambda \cdot T_{global}}{n_k + \lambda} \tag{11}$$

where n_k is the frequency of category $x_i^{(k)}$ and λ is a smoothing coefficient regulating smoothing strength.

By sorting the category features and leveraging the sorting results to create codes, CatBoost prevents the information leakage resulting from conventional coding techniques (Nguyen et al., 2024). Every category value $x_i^{(k)}$ can be represented as its sorting code:

$$Rank\left(x_i^{(k)}\right) = \frac{\sum_{j=1}^{m} I\left(rank\left(x_i^{(j)}\right) \le x_i^{(k)}\right)}{m} \tag{12}$$

where $rank(x_i^{(k)})$ is the ranked position of the category features among all the samples; I is an indicator function showing whether the rankings meet the requirement or not.

CatBoost does each round of decision tree training reducing the gradient loss. Assuming a $\hat{y}_t$ current model output and a y objective value, the gradient update formula for round t is:

$$g_t = \frac{\partial L\left(\hat{y}_t, y\right)}{\partial \hat{y}_t} \tag{13}$$

Usually the mean square error (MSE), or logarithmic loss function, L is the loss function; g_t is the gradient value.

CatBoost builds every tree in training using the best split point (Joshi et al., 2021). CatBoost determines whether to split at every turn by computing the gain of every feature. The formula determines the gain:

$$Gain(X) = \frac{1}{2}\left(\frac{\left(\sum_{i\in X} g_i\right)^2}{|X| + \lambda}\right) \tag{14}$$

where X is the subsample following split to avoid overfitting and g_i is the gradient of every sample.

CatBoost uses a symmetric tree construction with same node splitting rules at every tier (Chowdhury et al., 2024). This construction increases the training efficiency and helps each tree's subtrees to be more balanced. One can depict the tree's split in the following way:

$$Split(X) = \arg\max_k \left(Gain_k(X)\right) \tag{15}$$

where $Gain_k(X)$ is the gain of feature k over the collection of samples X.

CatBoost gradually adds the result of every tree to the prediction of the current model during the training phase; the weighted sum of all the outputs produces the final prediction:

$$\hat{y} = \sum_{t=1}^{T} w_t \cdot f_t(X) \tag{16}$$

where T is the total number of trees; w is the weight of the t^{th} tree; and $f_t(X)$ is the t^{th} tree's expected output.

Through minimising the loss function during training, CatBoost optimises the model parameters. The MSE is the often used loss function for the regression work:

$$L(\hat{y}, y) = \frac{1}{2}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{17}$$

where $\hat{y}_i$ is the expected value of the i^{th} sample; y_i is the real value; n is the overall sample count.

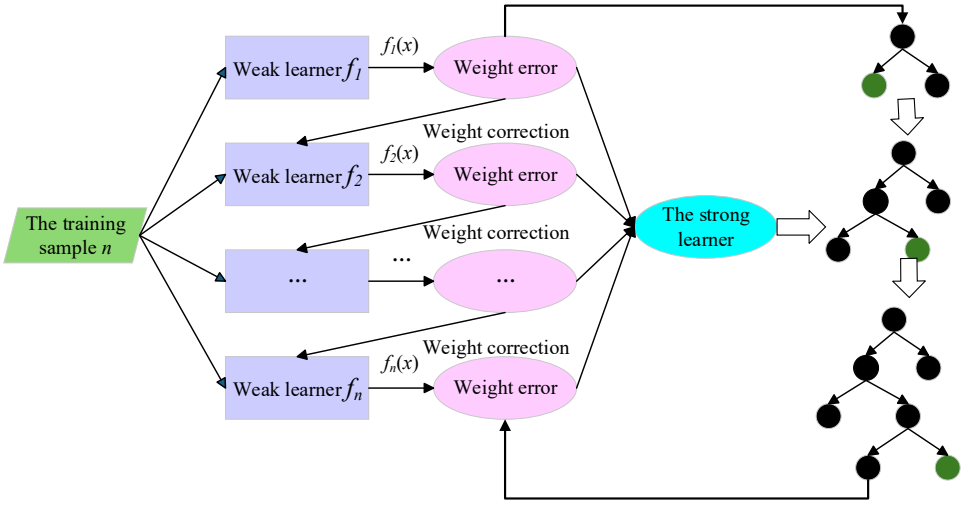
By effective encoding and optimisation of categorical features paired with the framework of GBT, CatBoost essentially increases the processing power on categorical data (So, 2024). By use of symmetric tree structure and gain computation, its special goal encoding and smoothing methods efficiently lower the danger of overfitting and increase the training efficiency of the model. Particularly in structured data processing, Catboost is a strong machine learning tool showing exceptional performance in many useful applications.

3 Machine learning based cyber security threat prediction model XG-CatSec

For the prediction of cybersecurity hazards in tourist hotels, the XG-CatSec model is meant to combine the benefits of two machine learning methods, XGBoost and Catboost,

in this chapter see Figure 2. Three core modules comprise the model: a prediction evaluation module, a module for data preparation, and a module for model training.

Figure 2 XG-CatSec model (see online version for colours)



3.1 Data preprocessing module

Data preprocessing is a key component of this module, particularly in cybersecurity threat prediction where the data typically consists of a great number of network traffic, user behaviours, attack events, and many other types of features, which need to be strictly cleaned and transformed to provide effective information for the model. First, this method generates features from unprocessed data, particularly from logs and network traffic collecting important information. Network traffic could include, for instance, timestamps, packet sizes, source IP addresses, destination ports, etc. For further model development, feature extraction will standardise and encode these data. This work uses normalisation for numerical feature x_i to guarantee that all numerical characteristics are on the same scale, therefore strengthening the stability of model training. The normalisation is obtained assuming that the maximum and minimum values of the original feature x_i are $x_{\max}$ and $x_{\min}$, respectively:

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{18}$$

Because of their different magnitude, the normalised feature x_i' ranges between [0, 1] helps to avoid the unfavourable influence of some features on model training. This work uses the one-hot encoding technique to convert every category feature-including source IP address and attack type-into a binary feature vector. This guarantees that the category data is handled without losing information and helps to prevent the model misreading the ordering relationships between categories.

3.2 Model training module

Through a fusion mechanism, these two algorithms complement each other's strengths while separately handle various kinds of features. XGBoost shines in handling numerical features; Catboost shines in handling categorical features. XGBoost and Catboost respectively apply distinct changes on characteristics in the stage of data preparation. They then cooperate via complementarity to optimise their prediction powers. XGBoost models numerical features specifically whereas Catboost concentrates on the optimisation of categorical features and by processing numerical and categorical data differently, they are able to respectively capture various patterns in the data.

The success of XGBoost and Catboost depends on their cooperation in this module. XGBoost's training mechanism is predicated on the following formula:

$$f_t(x) = f_{t-1}(x) + \eta \cdot \Delta f_t(x) \tag{19}$$

where $f_t(x)$ is the projected value of the model in the t^{th} iteration, $f_{t-1}(x)$ is the projected value in the previous round, η is the learning rate, and $\Delta f_t(x)$ is the improved value in the current round, so adjusting the weights of every tree by computing the second-order derivative. XGBoost may thus progressively adjust its prediction results for numerical features and optimise the prediction of cyber security hazards.

To further inform the model, the CatBoost model meanwhile concentrates on categorical features and converts categorical data into numerical features using a target coding method. CatBoost captures the link between category features and target labels by encoding every category by computing the target mean value of every category throughout the training phase. Assume a category characteristic $x_i^{(k)}$'s target coding to be $T(x_i^{(k)})$; this is computed as:

$$T\left(x_i^{(k)}\right) = \frac{\sum_{j=1}^{n} y_j \cdot \delta\left(x_i^{(j)} = x_i^{(k)}\right)}{\sum_{j=1}^{n} \delta\left(x_i^{(j)} = x_i^{(k)}\right) + \lambda} \tag{20}$$

where $\delta(x_i^{(j)} = x_i^{(k)})$ is an indicator function showing whether category $x_i^{(k)}$ exists in sample j or not; λ is a smoothing agent used to prevent low-frequency category overfitting. Particularly in situations when the category characteristics have high dimensionality or are unbalanced, CatBoost is able to handle them more effectively using this goal coding strategy.

Beyond separate modelling, both algorithms produce predictions that will be co-optimised by weighted fusion. The final cybersecurity threat prediction in the fusion phase combines XGBoost's forecasts with Catboost's The weighted fusion formula computes the final prediction result $\hat{y}$ assuming $\hat{y}_{XGBoost}$ as the prediction result of the XGBoost model and $\hat{y}_{CatBoost}$ as the prediction result of the CatBoost model:

$$\hat{y} = \alpha \cdot \hat{y}_{XGBoost} + (1-\alpha) \cdot \hat{y}_{CatBoost} \tag{21}$$

Usually tuned by cross-validation, where α is a weighting coefficient controlling the fusion ratio of XGBoost and Catboost prediction results. The fused prediction outcomes can concurrently exploit XGBoost's capabilities in numerical feature modelling and

Catboost's strengths in categorical feature modelling, so improving the prediction accuracy of the complete model.

This work also presents a model calibration phase in the fusion technique to improve the model performance even more. XGBoost and Catboost have distinct feature types respectively, hence, their outputs could differ somewhat. In this example, this work adjusts the outputs to be more consistent with the distribution of real labels using a basic post-processing technique, calibrated classifier. Assuming $\hat{y}_{raw}$ as the starting prediction derived by weighted fusion, this work can modify the outcomes using a calibration function:

$$\hat{y}_{calibrated} = g(\hat{y}_{raw}) \tag{22}$$

where $g(\hat{y}_{raw})$ is a calibration function learning from past data that translates the original predictions to a more accurate probability space. The calibrated results in cybersecurity threat prediction can help to increase the accuracy and resilience of the model and more precisely detect possible attackers.

3.3 Predictive assessment module

This work focuses on model evaluation utilising criteria including accuracy, precision, recall, F1 value, and AUC for the XG-CatSec model. The proportion of accurate predictions is measured by accuracy; performance of the model in positive and negative sample prediction is assessed by precision and recall; F1 value combines the balance of precision and recall; AUC is used to evaluate the classification capacity of the model.

Covering data preprocessing and the fusion of XGBoost and CatBoost, this chapter essentially describes the building process of the XG-CatSec model. By means of data cleaning, standardisation, feature selection, interactive feature generation, the data quality is guaranteed and a strong basis for the next model training is laid. The cooperation between XGBoost and Catboost balances each other in the creation of the fusion model to improve the general performance. This section offers a strong instrument for cybersecurity threat prediction by means of modular design and effective feature processing, hence improving the accuracy and predictive capacity of the model.

4 Experimental results and analyses

4.1 Experimental data

See Table 1 for the NSL-KDD (Network Security Laboratory KDD Cup 1999 Data Set) utilised in this investigation.

Although this dataset was not originally intended especially for the tourist and hospitality sector, it offers several kinds of cyber-attacks and sophisticated network traffic characteristics that can reasonably replicate attack threats in a general network environment. This study will change and map the dataset suitably to simulate cyber-attacks that may be faced by tourism and hotel management systems, online booking platforms, and IoT devices (e.g., smart door locks, room service devices), so adapting it to the task of cybersecurity threat prediction in the tourism and hospitality industry.

Table 1 NSL-KDD information

Feature	*Description*
Attack types	Denial of service (DoS), distributed denial of service (DDoS), user to root (U2R), remote to local (R2L)
Dataset size	Over 5,000 records
Record fields	Source IP, destination IP, source port, destination port, protocol type, etc.
Main application	Network intrusion detection, traffic analysis, threat detection
Data preprocessing	Data standardisation, missing value imputation, feature selection

4.2 Experimental procedure

This work will assess the performance difference between the XG-CatSec model and conventional machine learning algorithms in cybersecurity threat prediction by means of comparative experiments and investigate the contribution of every module in the XG-CatSec model to the general performance by ablation experiments.

First, a comparative experiment exists. With the intention of validating the XG-CatSec model in the cybersecurity threat prediction task against other common algorithms, this experiment initially compares the XG-CatSec model with many classic machine learning techniques. Using the identical training data and preprocessing methods – data normalisation and feature selection –XGBoost, Catboost, random forest, SVM, and LightGBM – each of which guarantees the fairness and validity of the experimental results – are the models chosen for comparison.

Under complicated assault patterns (e.g., DDoS and U2R), the XG-CatSec model clearly demonstrates evident benefits in key metrics like accuracy, precision, recall, and F1 value, according to the testing results; its prediction accuracy is thus far greater than that of other comparator models. XG-CatSec also shows improved resilience and can manage several attack kinds and variations in network traffic characteristics.

Figure 3 presents the outcomes of this comparison experiment.

The XG-CatSec model exhibits best results in all measures, particularly in accuracy (97.65%) and F1 value (96.43%), which are much better than the other models as the table makes clear. Though they both perform better, XGBoost and Catboost – with 94.35% and 93.82%, respectively – do not fare as well as the XG-CatSec model. Particularly in the identification of DDoS and U2R attack types, XG-CatSec shows improved accuracy and better recall, and can more precisely recognise these intricate assault patterns.

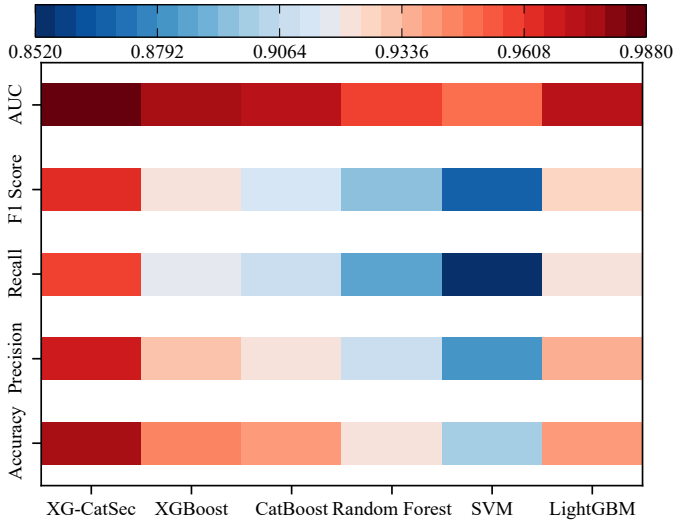
Furthermore, doing rather poorly with lower accuracy, recall, and F1 values is random forest and SVM, particularly in relation to challenging attacks like U2R. This implies that while the XG-CatSec model shows notable increases in prediction accuracy and robustness by virtue of its merger of the benefits of XGBoost and CatBoost, standard machine learning methods may find more difficulties with more sophisticated attacks.

This work validates the benefits of the XG-CatSec model in the task of cybersecurity threat prediction by means of comparison trials, particularly considering several attack kinds; its accuracy and robustness are really outstanding.

Ablation experiments come second here. Several experiments on the XG-CatSec model are carried out in this work with an eye toward analysing the contribution of every

component of the model to the general performance. Eliminating or replacing one module at a time helps one to better grasp the function of every algorithm and determine which components are essential to raise robustness and prediction accuracy.

Figure 3 Results of the comparison experiment (see online version for colours)



The first experiment ran removing the XGBoost component and training the model just with CatBoost. The aim of this experiment is to investigate if CatBoost can perform enough without the XGBoost module in the cybersecurity threat prediction in tourist hotels. Comparatively with the whole XG-CatSec model, the relevance of XGBoost in the model will be assessed based on the experiment outcomes.

The CatBoost component is then deleted and XGBoost is the sole tool used. This experiment is to investigate the contribution of the CatBoost module to the model performance and to evaluate the model performance when just XGBoost is used. One may see the benefit of CatBoost for identifying intricate attack patterns by means of comparison with the complete XG-CatSec model.

Furthermore discussed in this work is a feature ablation experiment. This experiment removes some elements from the dataset progressively in order to identify which ones are absolutely important for the model performance. For instance, one could eliminate network traffic characteristics such source IP, target IP, protocol type, etc. to observe if these influence the attack pattern identification capability. This helps one to confirm which features are most important in enhancing the predictive performance of the model and how much the model depends on certain aspects.

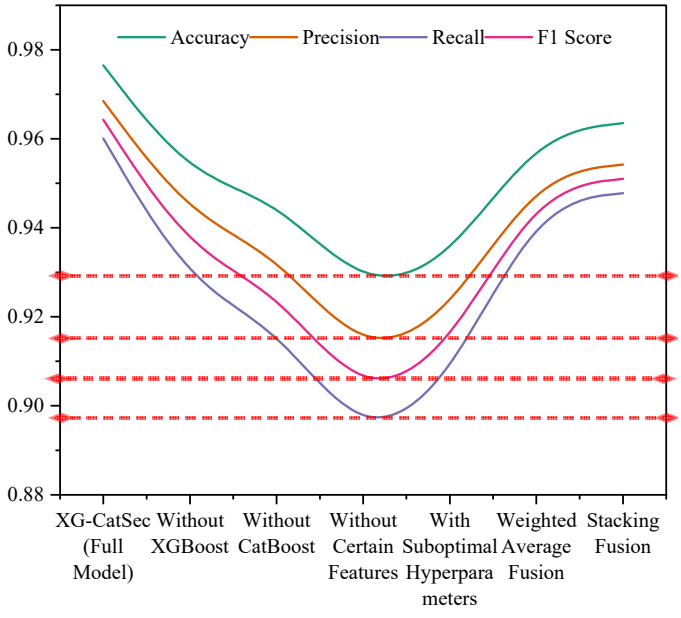
Another crucial ablation experiment is hyperparameter tweaking. In this work, it will tweak the hyperparameters of XGBoost and Catboost models, such the depth of the tree, the learning rate and the regularisation parameter, etc., and analyse the effects of different parameter settings on the model performance.

At last, this work additionally develops the fusion approach ablation experiment. The impacts of several fusion techniques on the XG-CatSec model will be investigated in this experiment. Different fusion techniques, such the weighted average approach and the stacking method, for instance, can be tested and the most appropriate fusion strategy for

the XG-CatSec model will be further verified by means of model performance comparison.

Figure 4 present some of the outcomes of this ablation experiment.

Figure 4 Results of the ablation experiment (see online version for colours)



The data clearly shows that on all measures the XG-CatSec model performs the best. The performance of the model declines somewhat following the removal of XGBoost or Catboost, particularly in accuracy, precision, and F1 value, so demonstrating the significance of both two algorithmic modules for the XG-CatSec model. Particularly when some characteristics are eliminated, the performance of the model declines, suggesting that the model effect is much influenced by the feature choosing. The hyperparameter tuning studies indicate, meanwhile, that suitable hyperparameter values can raise the model performance even more.

5 Conclusions

In this work, it presents a prediction model based on machine learning (XGBoost and Catboost) fusion of cybersecurity threats in tourism hotels, XG-Cat Sec. By means of a comparison with the conventional single-algorithm model, this study shows in the experimental phase XG-CatSec performs well on numerous indicators like prediction accuracy, precision, recall, and so on. rate; recall rate; several more measures. Particularly the synergistic effect of XGBoost and Catboost, the ablation experiment confirms the function of every module in the model and shows that the fusion of the two can considerably increase the robustness and prediction accuracy of the model.

The XG-CatSec model suggested in this work has certain restrictions even if it performs really well in the tests. First of all, there are certain restrictions in the dataset

applied for the trials. Though the NSL-KDD dataset is representative, it cannot totally cover all the real forms of network attack, particularly some emerging attack patterns. Second, the experimental setting of this work is more perfect and free of the impact of intricate network environment in practical implementation. Ultimately, even if XG-CatSec has been optimised with hyper-parameters, the choice of hyper-parameters still requires more careful adjustment depending on the situation in useful applications.

Future investigations in the following spheres is possible:

1 Dataset extension and diversity: Future studies should take into account widening the spectrum of cyberattacks by including cybersecurity data from several sources, therefore augmenting the variation of the dataset. Particularly for the identification of unknown assaults, this will enable the model to become more generalised in a changeable environment.

2 Model interpretability and transparency: e Although integrated models often show great predictive power, they are frequently complicated and challenging to understand. Future studies should concentrate on enhancing the interpretability of XG-CatSec models so that they not only correctly identify risks but also offer simple-to-understand decision support to enable security professionals make better choices and responses.

Declarations

All authors declare that they have no conflicts of interest.

References

Abdi, N., Albaseer, A. and Abdallah, M. (2024) 'The role of deep learning in advancing proactive cybersecurity measures for smart grid networks: a survey', *IEEE Internet of Things Journal*, Vol. 11, No. 9, pp.16398–16421.

Amiri, Z., Heidari, A., Navimipour, N.J., Unal, M. and Mousavi, A. (2024) 'Adventures in data analysis: a systematic review of Deep Learning techniques for pattern recognition in cyber-physical-social systems', *Multimedia Tools and Applications*, Vol. 83, No. 8, pp.22909–22973.

Asselman, A., Khaldi, M. and Aammou, S. (2023) 'Enhancing the prediction of student performance based on the machine learning XGBoost algorithm', *Interactive Learning Environments*, Vol. 31, No. 6, pp.3360–3379.

Azam, Z., Islam, M.M. and Huda, M.N. (2023) 'Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree', *IEEE Access*, Vol. 11, pp.80348–80391.

Bukowski, M., Kurek, J., Świderski, B. and Jegorowa, A. (2024) 'Custom loss functions in xgboost algorithm for enhanced critical error mitigation in drill-wear analysis of melamine-faced chipboard', *Sensors*, Vol. 24, No. 4, p.1092.

Chen, W., Wan, X., Ding, J. and Wang, T. (2024) 'Enhancing clay content estimation through hybrid CatBoost-GP with model class selection', *Transportation Geotechnics*, Vol. 45, p.101232.

Chowdhury, R., Chakraborty, T., Purkait, S. and Saha, B. (2024) 'SE2CURA-design and implementation of a robust ensemble learning based 2-tier intrusion detection system for real time traffic', *Multimedia Tools and Applications*, Vol. 83, No. 13, pp.38567–38609.

Chughtai, M.S., Bibi, I., Karim, S., Shah, S.W.A., Laghari, A.A. and Khan, A.A. (2024) 'Deep learning trends and future perspectives of web security and vulnerabilities', *Journal of High Speed Networks*, Vol. 30, No. 1, pp.115–146.

Dele-Afolabi, T., Jung, D., Ahmadipour, M., Hanim, M.A., Adeleke, A., Kandasamy, M. and Gunnasegaran, P. (2024) 'Jaya algorithm hybridized with extreme gradient boosting to predict the corrosion-induced mass loss of agro-waste based monolithic and Ni-reinforced porous alumina', *Journal of Materials Research and Technology*, Vol. 33, pp.5909–5921.

Demir, S. and Sahin, E.K. (2023) 'An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost', *Neural Computing and Applications*, Vol. 35, No. 4, pp.3173–3190.

Dong, Q., Su, Y., Xu, G., She, L. and Chang, Y. (2024) 'A Fast operation method for predicting stress in nonlinear boom structures based on RS-XGBoost-RF model', *Electronics*, Vol. 13, No. 14, p.2742.

Jha, K., Jain, A. and Srivastava, S. (2024) 'Analysis of human voice for speaker recognition: concepts and advancement', *J. Electr. Syst*., Vol. 20, No. 1s, pp.582–599.

Joshi, A., Saggar, P., Jain, R., Sharma, M., Gupta, D. and Khanna, A. (2021) 'CatBoost – an ensemble machine learning model for prediction and classification of student academic performance', *Advances in Data Science and Adaptive Analysis*, Vol. 13, No. 03n04, p.2141002.

Kandasamy, V. and Roseline, A.A. (2025) 'Harnessing advanced hybrid deep learning model for real-time detection and prevention of man-in-the-middle cyber attacks', *Scientific Reports*, Vol. 15, No. 1, p.1697.

Kimani, K., Oduol, V. and Langat, K. (2019) 'Cyber security challenges for IoT-based smart grid networks', *International Journal of Critical Infrastructure Protection*, Vol. 25, pp.36–49.

Li, J., An, X., Li, Q., Wang, C., Yu, H., Zhou, X. and Geng, Y-a. (2022) 'Application of XGBoost algorithm in the optimization of pollutant concentration', *Atmospheric Research*, Vol. 276, p.106238.

Lu, Y. and Da Xu, L. (2018) 'Internet of Things (IoT) cybersecurity research: a review of current research topics', *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp.2103–2115.

Mazhar, T., Irfan, H.M., Khan, S., Haq, I., Ullah, I., Iqbal, M. and Hamam, H. (2023) 'Analysis of cyber security attacks and its solutions for the smart grid using machine learning and blockchain methods', *Future Internet*, Vol. 15, No. 2, p.83.

Mohan, D.P. and Subathra, M. (2023) 'A comprehensive review of various machine learning techniques used in load forecasting', *Recent Advances in Electrical & Electronic Engineering*, Vol. 16, No. 3, pp.197–210.

Nguyen, T.M., Vo, H.H-P. and Yoo, M. (2024) 'Enhancing intrusion detection in wireless sensor networks using a GSWO-CatBoost approach', *Sensors*, Vol. 24, No. 11, p.3339.

Sarker, I.H. (2021) 'Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective', *SN Computer Science*, Vol. 2, No. 3, p.154.

Sarker, I.H. (2023) 'Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects', *Annals of Data Science*, Vol. 10, No. 6, pp.1473–1498.

Shaik, N.B., Jongkittinarukorn, K. and Bingi, K. (2024) 'XGBoost based enhanced predictive model for handling missing input parameters: a case study on gas turbine', *Case Studies in Chemical and Environmental Engineering*, Vol. 10, p.100775.

So, B. (2024) 'Enhanced gradient boosting for zero-inflated insurance claims and comparative analysis of CatBoost, XGBoost, and LightGBM', *Scandinavian Actuarial Journal*, Vol. 2024, No. 10, pp.1013–1035.

Zhang, L. and Jánošík, D. (2024) 'Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches', *Expert Systems with Applications*, Vol. 241, p.122686.