UNIVERSITI TEKNOLOGI MARA

CSP760

PREDICTING VULNERABILITY SUSCEPTIBILITY IN MALAYSIAN BANK USING SUPERVISED MACHINE LEARNING

STUDENT NOR ADANI BINTI KAMAL MOHAMAD NASIR (2024782087)

SUPERVISOR DR SITI ARPAH BINTI AHMAD

Date: 21 December 2025

TABLE OF CONTENT

RELATED WORKS

No	Title	Main Issue	Objective	Dataset	Algorithms	Solution
1	A cyber risk prediction model using common vulnerabilities and exposures (Negahdari Kia et al., 2023)	Predicting cyber risks using CVE data with supervised ML models	Eliminate expert bias and predict cyber risks through ML	CVE Database with topic mapping	Random Forest, Time Series Analysis	Generate a time-series risk prediction model, CyRiPred
2	A Hybrid Machine Learning System for Vulnerability Detection in Web Applications (Oliveira, 2023)	Hybrid ML approach for detecting vulnerabilities in web applications	Develop a hybrid ML model combining NLP and anomaly detection	Software Assurance Reference Database (SARD)	OCSVM, Random Forest, Logistic Regression	Propose a hybrid model integrating NLP and ML
3	A Vulnerability Analysis and Prediction Framework (Williams et al., 2020)	Predicting and analyzing vulnerability evolution over time	Develop a predictive framework for vulnerability trends	National Vulnerability Database (NVD)	Deep Neural Networks, Regression	Use topic modelling and storytelling techniques for vulnerability forecasting
4	Comprehensive Survey of different Machine Learning Algorithms used for Software Defect Prediction (K et al., 2022)	Addressing software defects using various ML techniques	Survey and analyze different ML algorithms for software defect prediction	PROMISE Repository, Software defect datasets	Random Forest, Naive Bayes, SVM, Decision Tree, ANN, K-Means Clustering	Comprehensive evaluation of supervised and unsupervised ML methods for defect prediction
5	Time series forecast modelling of vulnerabilities in the android operating system using ARIMA and deep learning methods (Gencer & Başçiftçi, 2021)	Forecasting future vulnerabilities in Android OS	Use time series and deep learning for vulnerability prediction	National Vulnerability Database (NVD) filtered for Android	ARIMA, LSTM, CNN	Apply deep learning models to predict Android vulnerabilities
6	Integrating Machine Learning for Sustaining Cybersecurity in Digital Banks (Asmar & Alia Tuqan, 2024)	Cybersecurity threats in digital banking and the need for ML-based solutions	Strengthen cybersecurity defenses using ML in digital banking	Literature review, cybersecurity threat reports	SVM, RNN, HMM, LOF	Develop an ML-driven cybersecurity framework for digital banks
7	Predicting Vulnerability Type in CVE Database with ML Classifiers (Yosifova et al., 2021)	Automating the classification of vulnerability types in CVE database	Enhance automated classification of CVE vulnerability types	CVE Database	Linear SVM, Naive Bayes, Random Forest	Train ML classifiers for improved CVE classification
8	Predicting Vulnerability Susceptibility in Malaysian Bank using Supervised Machine Learning	Current VA tools in Malaysian banks are reactive, lack predictive insights	Develop a machine learning model to predict cyberattack susceptibility & improve remediation efficiency.	Kaggle, NVD, ExploitDB, Tenable	Random Forest, Neural Networks, Regression	Implement an AI-driven system to analyze VA data, forecast emerging threats, and provide real-time vulnerability insights for proactive mitigation.

Top5_Random Forest - Results

```python
def top_k_accuracy(y_true, y_proba, k=5):
    y_true_arr = np.array(y_true)
    correct = 0
    for i in range(len(y_true_arr)):
        topk_idx = np.argsort(y_proba[i])[::-1][:k]
        if y_true_arr[i] in topk_idx:
            correct += 1
    return correct / len(y_true_arr)

top5_acc = top_k_accuracy(y_test_rf, y_proba, k=5)
top5_acc
```

```
0.972369234998969
```

```python
# Compute Top-5 accuracy for NN classifier
import numpy as np

y_proba_nnc = nnc.predict_proba(X_test_cls)

def top_k_accuracy(y_true, y_proba, k=5):
    y_true_arr = np.array(y_true)
    correct = 0
    for i in range(len(y_true_arr)):
        topk_idx = np.argsort(y_proba[i])[::-1][:k]
        if y_true_arr[i] in topk_idx:
            correct += 1
    return correct / len(y_true_arr)

top5_acc_nnc = top_k_accuracy(y_test_rf, y_proba_nnc, k=5)
top5_acc_nnc
```

```
0.9872156161935528
```

Top5_Neural Network Classifier - Results

```
              precision    recall  f1-score   support

           0       0.06      0.38      0.11        13
           1       0.29      0.48      0.36       814
           2       0.04      0.36      0.08        86
           3       0.50      0.71      0.59       575
           4       0.89      0.56      0.69     10672
           5       0.06      0.60      0.11       184
           6       0.37      0.30      0.33      2022
           7       0.03      0.36      0.05        14
           8       0.01      0.23      0.02        30
           9       0.27      0.77      0.39       139

    accuracy                           0.53     14549
   macro avg       0.25      0.48      0.27     14549
weighted avg       0.74      0.53      0.60     14549
```

Random Forest - Results

1. Neural Network Classifier is better at Top-1
- It strongly favors the dominant class (class 4)
- It ignores rare classes (many 0.00 recalls)

2. Both models are excellent at Top-5

RF: 97.2%

NN: 98.7%

3. Neural Network is more biased
- classes 0,2,5,7,8 → all zero recall
- dangerous for security
- looks good numerically, but misses **rare attack types**

4. Random Forest is more balanced
- Lower Top-1, but
- better coverage
- more interpretable
- more stable for rare attacks

Model	Top-1 Accuracy	Top-5 Accuracy	Desc
Random Forest	**~0.53**	**0.972**	Balanced, interpretable
Neural Network (Classifier)	**0.76**	**0.987**	Biased toward dominant class

Neural Network Classifier - Results

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        13
           1       0.66      0.17      0.27       814
           2       0.00      0.00      0.00        86
           3       0.73      0.36      0.48       575
           4       0.77      0.99      0.86     10672
           5       0.00      0.00      0.00       184
           6       0.67      0.08      0.14      2022
           7       0.00      0.00      0.00        14
           8       0.00      0.00      0.00        30
           9       0.91      0.50      0.65       139

    accuracy                           0.76     14549
   macro avg       0.37      0.21      0.24     14549
weighted avg       0.73      0.76      0.69     14549
```

AdaniKamal/**Predicting-CyberAttack**

Its about my project

PROTOTYPE

اونيۏرسيتي تيکنولوݢي مارا
UNIVERSITI TEKNOLOGI MARA

CSP760

THANK YOU