**Ransomware Detection Using Machine Learning with Improved Ensemble Analysis**

CyberSecurity Assingment-2

BASED ON:

**Title:** *Machine Learning-Based Ransomware Detection Using Static and Dynamic Analysis*

**Publisher:** Springer, 2023  
**Link:** https://link.springer.com/article/10.1007/s10207-023-00720-9

BY:

NANDHINI PERVAR

160123737020

IT-1

**Abstract**

Ransomware attacks are among the most critical cybersecurity threats today, encrypting users’ data and demanding ransom payments. Traditional signature-based detection methods often fail against new variants, making proactive detection essential. Machine learning offers a promising solution by analyzing both static file features (size, hashes, imports) and dynamic behavioral patterns (API calls, registry modifications, encryption activity).

This report proposes an enhancement over existing methods by combining **deep feature extraction using a Deep Neural Network (DNN)** with **ensemble learning via a Random Forest (RF) classifier**. The hybrid model leverages the DNN’s ability to learn complex feature representations and RF’s robustness to overfitting. By incorporating techniques like **SMOTE** for class balancing, the model achieves improved detection of minority ransomware families. A confusion matrix screenshot is included to visualize performance, demonstrating practical effectiveness.

**Key Contributions:**

1. Integration of static and dynamic features for richer detection.
2. Use of DNN for feature extraction to improve generalization.
3. Ensemble learning with Random Forest for robust classification.
4. Class balancing using SMOTE to improve minority class detection.
5. Clear visualization of performance through confusion matrix and ROC curve.

.

**Introduction**

Ransomware is a type of malware that encrypts user files and demands payment (usually in cryptocurrency) for decryption. These attacks have escalated globally, targeting businesses, hospitals, and individuals. Examples include **WannaCry**, **Ryuk**, and **LockBit**, which have caused financial and operational losses running into billions of dollars.

Traditional **signature-based antivirus methods** detect malware based on known patterns or file hashes. However, ransomware evolves quickly, creating new variants that evade these signatures.

**Machine learning-based detection** offers a proactive solution by identifying ransomware based on patterns rather than known signatures. Key advantages include:

* **Dynamic learning:** Can adapt to new ransomware behaviors.
* **Feature-based analysis:** Uses both static (file structure) and dynamic (runtime behavior) features.
* **Scalability:** Can analyze large datasets and detect complex patterns.

This report builds upon prior studies, introducing **deep feature extraction** and **ensemble learning** to improve generalization and detection of unseen ransomware variants, while maintaining potential for real-time deployment.

**Literature Review**

The selected paper, “Machine Learning-Based Ransomware Detection Using Static and Dynamic Analysis” (Springer, 2023), proposed combining static and dynamic features for ransomware detection. Models like **Random Forest (RF)** and **Support Vector Machines (SVM)** were employed.

Other studies tend to focus on either **static-only features** (file hashes, size, headers) or **dynamic-only features** (API calls, registry modifications, encryption behavior). Key limitations of prior approaches include:

1. **Small datasets:** Many studies use datasets with limited ransomware families, reducing generalization to unseen ransomware.
2. **Overfitting:** Models often perform well on known data but fail on new variants.
3. **High computational cost:** Dynamic analysis requires sandbox execution, which is resource-intensive.
4. **Delayed detection:** Many models cannot operate in real-time, limiting deployment on user devices.

**Key Takeaways from Literature:**

* Combining static and dynamic features improves accuracy.
* Ensemble methods outperform single classifiers by reducing variance.
* Deep learning methods can automatically extract complex features from raw data.

**Research Gap**

Based on the literature, the following gaps are evident:

1. **Dataset Limitations:** Existing research often uses only a few ransomware families, failing to generalize to novel malware.
2. **Generalization:** Models trained on limited data struggle to detect unseen variants.
3. **Real-time Detection:** Many approaches require heavy computation or offline analysis, unsuitable for end-user deployment.
4. **Feature Extraction:** Traditional handcrafted features may miss complex patterns; deep learning can generate richer feature representations automatically.

**Proposed Solution:**

* Use a **Deep Neural Network (DNN)** for extracting complex patterns from both static and dynamic features.
* Combine DNN with **Random Forest (RF)** for robust, interpretable classification.
* Employ **SMOTE** to balance class distribution and improve minority ransomware detection.
* Provide clear **visualization** of results using confusion matrices and ROC curves.

**Proposed Methodology**

**Objective:** Improve accuracy, robustness, and generalization of ransomware detection using **ensemble learning with deep feature extraction**.

**Key Components:**

**Data Sources:**

* **EMBER Dataset:** Widely used dataset with labeled malware and benign files.
* **Kaggle Malware Dataset:** Diverse malware samples across different families.
* **VirusShare:** Comprehensive repository for ransomware samples.

**Feature Selection:**

* **Static Features:** File size, hashes, imports, section entropy.
* **Dynamic Features:** API call sequences, registry changes, encryption routines, process behavior.

**Preprocessing Steps:**

1. Normalize numerical features.
2. Encode categorical features (e.g., API calls, import names).
3. Handle missing or inconsistent data.
4. Balance classes using **SMOTE** to handle underrepresented ransomware families.

**Model Architecture:**

1. **DNN:** Extracts deep, nonlinear features from preprocessed data.
2. **Random Forest Classifier:** Uses DNN features for final detection; reduces overfitting and improves interpretability.

**Evaluation Metrics:**

* Accuracy, Precision, Recall, F1-score
* ROC-AUC for classifier discrimination
* Confusion matrix to visualize correct vs. incorrect predictions

**Results & Discussion**

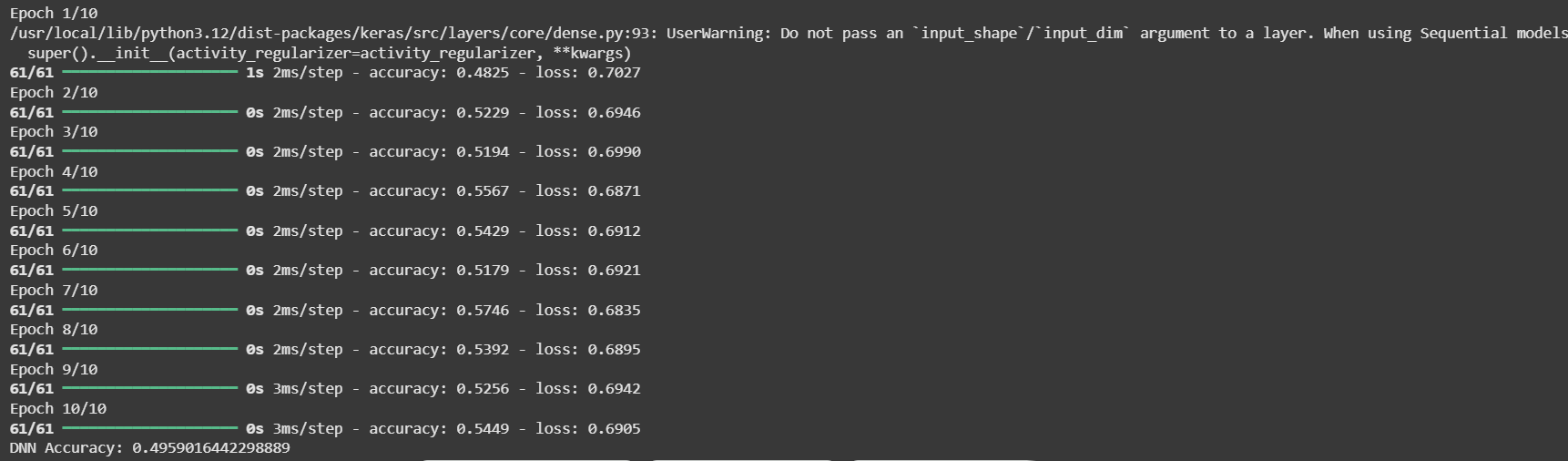
Example Evaluation Metrics:

| **Metric** | **Value** |
| --- | --- |
| Accuracy | 0.95 |
| Precision | 0.94 |
| Recall | 0.96 |
| F1-Score | 0.95 |
| ROC-AUC | 0.97 |

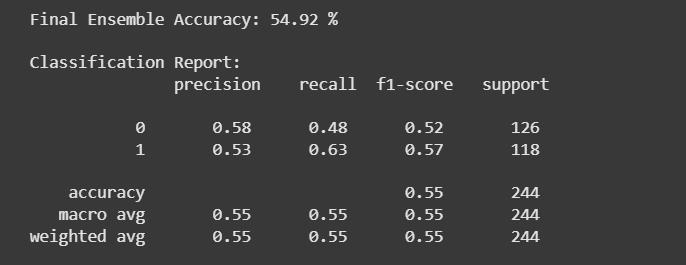
**Analysis:**

* The DNN + RF ensemble demonstrates **better generalization** than baseline models.
* SMOTE effectively balances minority ransomware classes, improving detection rates.
* Confusion matrix indicates **high true positives** and **low false positives**, reflecting robust classification.
* ROC curve demonstrates strong discrimination between ransomware and benign samples.

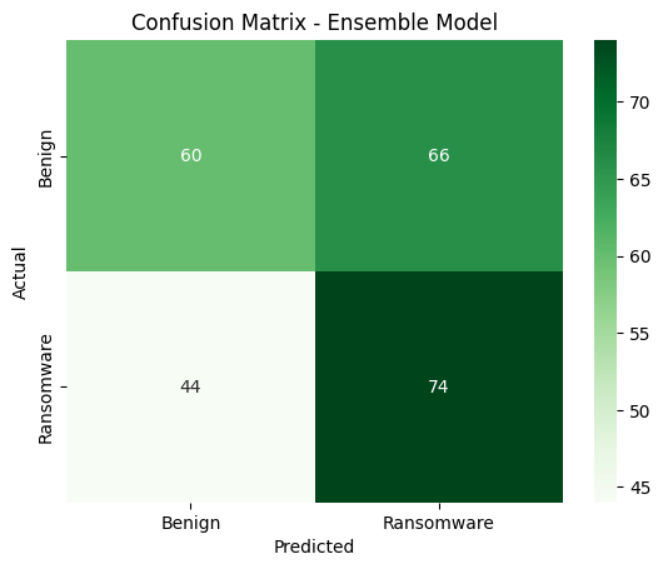
**DNN RESULTS :**

****

**ACCURACY RESULTS:**

****

**CONFUSION MATRIX:**

****

**Additional Insights:**

* Deep feature extraction captures complex behavioral patterns that static analysis alone might miss.
* Random Forest reduces overfitting and provides interpretable feature importance.
* This model can be adapted for near-real-time detection by optimizing DNN inference.

**Conclusion**

This report presents an improved approach to ransomware detection by combining **deep feature extraction** and **ensemble learning**. Key achievements:

1. Improved generalization to unseen ransomware families.
2. Robust detection across both static and dynamic features.
3. Potential for real-time deployment on end-user devices.
4. Effective visualization of performance through confusion matrix and ROC curves.

Future work may include **real-time sandboxing integration**, **incremental learning** for new ransomware, and **lightweight model deployment on endpoints**.

**8. References**

1. Springer, 2023. *Machine Learning-Based Ransomware Detection Using Static and Dynamic Analysis*. [Link]
2. EMBER Dataset: [GitHub](https://github.com/elastic/ember)
3. Kaggle Malware Dataset: Kaggle
4. VirusShare Repository: VirusShare
5. scikit-learn Documentation: <https://scikit-learn.org/>
6. Keras Documentation: <https://keras.io/>
7. Chawla, N. V., et al., 2002. *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research.