information

MDPI

Article

Novel Ransomware Detection Exploiting Uncertainty and Calibration Quality Measures Using Deep Learning

Mazen Gazzan [1,2,*] and Frederick T. Sheldon [1,*]

[1] Department of Computer Science, College of Engineering, University of Idaho, Moscow, ID 83844, USA
[2] Department of Information Systems, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia
[*] Correspondence: gazz6545@vandals.uidaho.edu or mzgazzan@nu.edu.sa (M.G.); sheldon@uidaho.edu (F.T.S.)

Abstract: Ransomware poses a significant threat by encrypting files or systems demanding a ransom be paid. Early detection is essential to mitigate its impact. This paper presents an Uncertainty-Aware Dynamic Early Stopping (UA-DES) technique for optimizing Deep Belief Networks (DBNs) in ransomware detection. UA-DES leverages Bayesian methods, dropout techniques, and an active learning framework to dynamically adjust the number of epochs during the training of the detection model, preventing overfitting while enhancing model accuracy and reliability. Our solution takes a set of Application Programming Interfaces (APIs), representing ransomware behavior as input we call "UA-DES-DBN". The method incorporates uncertainty and calibration quality measures, optimizing the training process for better more accurate ransomware detection. Experiments demonstrate the effectiveness of UA-DES-DBN compared to more conventional models. The proposed model improved accuracy from 94% to 98% across various input sizes, surpassing other models. UA-DES-DBN also decreased the false positive rate from 0.18 to 0.10, making it more useful in real-world cybersecurity applications.

Keywords: ransomware; early detection; deep learning; early stopping mechanisms; dynamic bayesian; deep belief network

check for updates

Citation: Gazzan, M.; Sheldon, F.T. Novel Ransomware Detection Exploiting Uncertainty and Calibration Quality Measures Using Deep Learning. *Information* **2024**, *15*, 262. https://doi.org/10.3390/info15050262

Academic Editors: Nelly Leligou, Theodore Zahariadis, Panagiotis Trakadas and Panagiotis A. Karkazis

Received: 14 March 2024
Revised: 1 May 2024
Accepted: 2 May 2024
Published: 5 May 2024

1. Introduction

In the digital age, devices are typically connected, facilitating the exchange of data and making it easy for users to communicate. Such connectivity introduces various security risks to user and business data [1]. Among these risks are the malware attacks utilized by attackers to steal data, disrupt operations, or hijack systems causing operational interruptions as well as impacts affecting reputation and compliance [2]. Ransomware is one type of malware that attackers use to lock user data using the operating system's own cryptographic utilities [3,4]. Ransomware presents a significant threat to organizations and individuals, as it encrypts files rendering computer systems inoperable. A decryption key may be provided after a ransom is paid [5,6], but not always. The irreversible effect of such an attack can be devastating due to the disruption in services and especially because, in some cases, after paying the ransom, the decryption key provided is useless [7] in restoring operations. Detecting ransomware early is crucial to minimizing its impact. Several studies [7–9] emphasize the importance of timely detection in reducing the damage caused by ransomware attacks. Urooj et al. [10] highlight the significance of early detection by proposing a machine learning-based model for detecting ransomware before the encryption phase takes place. Additionally, [11] suggests monitoring external server connections to identify ransomware activities early on, thereby preventing the completion of the encryption processes. Bold et al. [12] stress the importance of intelligent early detection in reducing false negatives, emphasizing the advantages of promptly identifying and eradicating ransomware. Thus, early detection of ransomware through advanced detection techniques is essential to prevent data loss, financial harm, and operational disruptions resulting from ransomware attacks.

Conventional deep-learning methods have been used in the detection of malware and ransomware. Al-Garadi et al. [13] discussed advancements in machine and deep learning methods for Internet of Things (IoT) security, showcasing the practical basis for the use of machine intelligence. Shaukat et al. [14] provided insights into the performance and challenges of machine learning techniques in cybersecurity, emphasizing the critical role of deep learning models in enhancing security measures. Regarding malware detection, Liu et al. [15] conducted a literature review on deep learning for Android malware defenses, highlighting trending research focused on deep learning techniques to combat Android malware attacks. Additionally, Uysal et al. [16] conducted a survey on data-driven malware detection, emphasizing the importance of continuous learning and explainability through visualization thus improving malware detection performance. Urooj et al. [17] explored ransomware detection using dynamic analysis and machine learning, demonstrating the effectiveness of deep learning methodologies in identifying and mitigating ransomware attacks. Furthermore, Shemitha and Dhas [18] developed a deep belief network model for ransomware detection, emphasizing the significance of optimal feature selection for enhancing the detection performance of deep learning models.

Deep learning models have gained significant attention in the context of cybersecurity using the various models being developed and tested for misuse and anomaly detection and to prevent or deter different types of intrusions [19]. The rise of deep learning has improved the conventional signature and specification-based detection solutions, making these crucial tools for protecting systems and networks against cyberattacks [20]. Deep learning is recognized for its ability to rapidly analyze data streams as a basis to actively deploy and adjust countermeasures for improved detection of malware attacks [21,22]. Furthermore, the limitations of conventional approaches have led to the proposal of hierarchical deep learning systems for analyzing content and behavioral features to attack patterns and intrusions [23]. In the context of early stopping mechanisms, it is crucial to consider the potential of deep learning models to overfit the training data leading to reduced generalization performance (A common metric of inference accuracy, or error, for input data of a trained neural network model, and not for the training data). This is where early stopping mechanisms can play an important role in preventing overfitting. By monitoring a model's performance on a separate validation dataset and thereby stopping the training process when the performance begins to degrade [24]. In this paper, we show that the use of early stopping in deep learning-based malware detection can improve the generalization performance of deep learning models by enhancing their accuracy specifically against ransomware attacks.

The existing methods used for early stopping of deep learning model training encompass a range of strategies to prevent overfitting and enhance generalization performance. In their study, Cho et al. [25] emphasized the utilization of early stopping to prevent performance degradation from excessive training, highlighting its significance in hyperparameter tuning of deep neural networks. Dorka et al. [26] highlighted the popularity of early stopping based on validation set performance to achieve a balance between underfitting and overfitting in supervised learning. Rezaeezade and Batina [27] discussed the use of early stopping as one of the techniques to combat overfitting in deep learning-based side-channel analysis, emphasizing its role as a regularizer. The study conducted by Moodley et al. [28] demonstrates the efficiency of early stopping in non-degenerate ghost imaging, showing potential for real-time implementation across various scenarios. Kaandorp et al. [29] explored the impact of early stopping criteria on intravoxel incoherent motion modeling, emphasizing its role in ensuring adequate convergence during training. Dossa et al. [30] conducted an empirical investigation into early stopping optimizations in proximal policy optimization, highlighting the practical implications of reinforcement learning. Choi and Lee [31] have studied a method that exploits all samples in low-resource sentence classification through methods that utilize early stopping and initialization parameters, which demonstrate their versatility across different applications. Moreover, two other studies, Wang et al. [32], and Li et al. [33], have utilized early stopping in the deep image prior and single-instance deep generative priors, respectively. Their work further indicates their potential utility in diverse deep-learning contexts. Furthermore, Dai et al. [34] have proposed DIPDefend, a defense mechanism against adversarial examples emphasizing the role of early stopping toward achieving robustness against attacks as well as creating augmented resilience.

Conventional measures such as validation accuracy, which are commonly used to determine when to stop training, may be influenced by data skew, which may result in underfitting. Such measures, if they do not accurately capture significant improvements that occur later in the training process, are destined to fail when faced with more difficult examples. Thus, the implementation of adaptive learning rates, which are a widely used optimization technique, introduces dynamic adjustments that static stopping criteria usually do not consider. This may result in missed opportunities to achieve even better generalization performance. Such constraints underscore the necessity for more advanced methods in terminating the training process for deep learning models. Studies utilizing dynamic stopping have shown they are adaptable to changes in training dynamics, such as variations in data complexity and fluctuations in learning rate.

While current dynamic stopping criteria have certain benefits compared to static metrics, the inclusion of uncertainty and variability holds the potential for additional progress. Current point estimates of performance frequently fail to consider these factors, which may result in incorrect decisions regarding when to stop. Bayesian methodologies, dropout techniques, calibration methods, and active learning frameworks present promising opportunities for incorporating uncertainty and variability. This can result in (i) more precise stopping decisions, (ii) training that requires fewer data, and (iii) improved generalization from the model. Considering these factors will likely facilitate the realization of more dependable dynamic stopping techniques that mitigate overfitting and enhance the precision of deep learning models. To this end, this paper is devoted to demonstrating a dynamic early-stopping method for DBN-based ransomware detection, which overcomes the limitations of existing solutions. The contribution of this paper is three-fold.

1. We proposed and developed an Uncertainty-Aware Dynamic Early Stopping (UA-DES) technique to optimize the training process of Deep Belief Networks (DBNs).
2. We integrated the UA-DES into a DBN-based ransomware detection model to prevent both underfitting and overfitting, for more accurate detection to test our theory.
3. We evaluated the performance of the proposed technique against existing solutions.

The lack of available ransomware data poses a significant obstacle to training deep learning models. Primarily, this is because of the high likelihood of overfitting, a problem that is less common in other domains with access to larger datasets such as computer vision. To address this issue, we utilize various strategies to reduce the occurrence of overfitting. Advanced regularization techniques, such as dropout and data augmentation, aim to increase the variety of small training sets. Meanwhile, the UA-DES mechanism monitors performance to stop training early if overfitting is detected. Additionally, our active learning framework enhances data utilization by selectively training on the most informative samples, toward enhancing model performance despite the limitations of the available data. These integrated methodologies are designed to optimize knowledge acquisition from a limited dataset in the context of ransomware detection.

The rest of this paper is organized as follows. The related works are explained in Section 2. The methodology is described in Section 3. The results and discussion are in Section 4. Conclusive remarks are found last in Section 5.

2. Related Studies

Deep learning has been identified as a promising approach for the early detection of ransomware, offering enhanced capabilities in identifying and mitigating evolving cyber threats. Research by Urooj et al. [17] has demonstrated the effectiveness of dynamic analysis and machine learning in ransomware detection, highlighting the potential of deep learning models in strengthening cybersecurity measures. Almomani et al. [35] have pointed out the limitations of traditional malware detection methods in combating ransomware. They have stressed the importance of utilizing deep learning techniques to address challenges posed by highly imbalanced data in ransomware detection scenarios. These insights collectively highlight the crucial role of deep learning in improving the early detection of ransomware, enabling proactive measures to protect digital assets against malicious attacks. Moreover, the integration of deep learning methodologies in ransomware detection has shown promising results in enhancing detection accuracy and efficiency. Sharmeen et al. [36] introduced a framework that incorporates behavioral and anomaly detection schemes powered by deep learning to enhance the accuracy of ransomware detection in the early stages before encryption occurs. The study by Fernando and Komninos [37] explored the evolution of ransomware detection using machine learning and deep learning techniques, weighing the critical need for early detection mechanisms to mitigate the destructive impacts of ransomware attacks. These findings underscore the significance of leveraging deep learning algorithms for ransomware detection, enabling proactive identification and mitigation of threats before they escalate, thereby enhancing the overall cybersecurity posture for organizations and individuals.

Early stopping mechanisms are crucial in training deep learning models for malware and ransomware detection, ensuring optimal model performance and preventing overfitting. Studies such as Kim et al. [38] have highlighted the significance of multimodal deep learning methods in malware detection, emphasizing the need for effective early stopping strategies to enhance model training efficiency. The utilization of ensemble models, as demonstrated by Hemalatha et al. [39], underscores the importance of combining multiple deep-learning models for robust malware detection. Such early stopping mechanisms can aid in optimizing the ensemble model's performance by preventing unnecessary training iterations and improving convergence. In the context of malware detection based on deep neural network (DNN) models, researchers like Du et al. [40] have shown a growing interest in autonomic methods utilizing deep learning for malware detection. Early stopping mechanisms are essential in training DNN models effectively, ensuring that the models do not overfit the training data and can effectively generalize toward robust detection of new and evolving malware threats. The study by [41] emphasizes the need for utilizing deep sequence learning models like Long Short-Term Memory (LSTM) models for malware detection based on network traffic. In all of these works, the emerging trend highlights the use of early stopping criteria to optimize the training process and enhance model generalization performance.

Deep Belief Networks (DBNs) have emerged as a powerful tool in the realm of cybersecurity, particularly in the detection of malware and ransomware. Various studies have explored the efficacy of DBNs in enhancing detection accuracy and robustness against evolving cyber threats. Al-Garadi et al. [13] showed significant improvements in malware detection accuracy by utilizing a DBN-based model compared to traditional machine learning algorithms. Similarly, Duhayyim et al. [42] employed the Artificial Algae Optimization Algorithm with Optimal Deep Belief Network (AAA-ODBN) technique, by integrating DBNs for ransomware detection in IoT environments, showcasing the versatility of DBNs in addressing diverse cybersecurity challenges. Urooj et al. [17] focused on ransomware detection using DBNs, emphasizing the effectiveness of DBNs in hardware-based security solutions. Moreover, DBNs have been successfully applied in detecting malware attacks, as highlighted by [43], who evidenced the utility of DBNs in detecting malware. Additionally, DBNs have been utilized in extracting invariant representations of malware behavior, as evidenced by [16], exposing the adaptability of DBNs to capture complex malware behavior patterns. Moreover, DBNs have been demonstrated to be instrumental in Android malware detection with studies by [44,45] which demonstrated the efficacy of DBNs in processing static features and enhancing detection accuracy. The application of DBNs in cybersecurity extends to the detection of intrusions and malware, as highlighted by [46], emphasizing the broad utility of DBNs in safeguarding systems against cyber threats.

As a graphical model, the DBN has multiple layers as defined by J. Qiu, et. al. [47], that generate data. Therein, the top two layers form associative memory and the lower layers create a belief network for probabilistic estimation of variable states and their connections to input data. DBNs stand out from other neural networks due to their unique architecture and learning approach, utilizing stacked Restricted Boltzmann Machines (RBMs) for pre-training to capture data representations and joint probability distributions, followed by possible supervised fine-tuning. The "belief" in DBNs, a quantitative likelihood, helps identify hidden data patterns, making them particularly adept at tasks like ransomware detection which for discerning the subtle signs of malicious activity is essential.

The DBN uses several hyperparameters that can be optimized for a specific application. The values of these parameters are adjusted at the beginning of the model training. Among these parameters, batch size, L2 regularization, momentum, dropout, and learning rate are the most effective for improving generalization performance. Within a DBN, the term "batch size" pertains to the number of training examples that are processed during a single forward/backward pass (i.e., epoch).

- Batch size has an impact on both the speed and stability of the learning process.
- L2 regularization is a technique employed to mitigate overfitting by imposing a penalty on the size of the model's parameters.
- Momentum is a parameter that aids in the acceleration of gradients in the appropriate direction, thereby facilitating the smoothing of updates during training.
- Dropout is a regularization technique that involves randomly excluding certain neurons during the training process. This helps to mitigate overfitting by reducing the model's reliance on the precise weights of individual neurons.
- The learning rate is a crucial factor during the iterative process of moving towards the minimum of a loss function. Learning rate determines the size of the step taken at each iteration and plays a key role toward efficiently converging to the best, so-called "best" solution.

3. The Methodology

This section provides a detailed account of the methodology employed to create the proposed early-stopping method for the DBN-based ransomware detection model. We begin by explaining the design of the suggested dynamic early stopping technique, and subsequently explore its integration within the detection model. An explanation of the dataset and technique for training the model is provided.

3.1. Uncertainty-Aware Dynamic Early Stopping (UA-DES)

The UA-DES technique is a method for improving the training process of the DBN model for ransomware detection. The process starts by utilizing the Bayesian method to create a model of the posterior distribution of performance metrics. This enables measurement of the level of uncertainty associated with these metrics described above in the previous section. The dropout technique is utilized as a Bayesian approximation that models the posterior distribution which facilitates the computation of predictive uncertainty. Temperature Scaling is used as a calibration technique to enhance the accuracy of probability estimates, ensuring that they are dependable and accurately represent the desired performance. A framework for active learning is incorporated to prioritize training on samples that are anticipated to yield the highest information gain, thereby improving the efficiency of the model's learning process. The process concludes with a dynamic stopping criterion that integrates performance improvement thresholds with measures of uncertainty and calibration quality. These criteria guarantee that training ceases not only when there are minimal improvements in performance, but also when the model attains a satisfactory level of uncertainty and calibration. By ensuring that the model is both precise and dependable, our proposed UA-DES model enhances the generalization and accuracy of DBNs, thereby increasing their effectiveness in detecting ransomware attacks. Figure 1 shows the general structure and process implemented by the UA-DES.

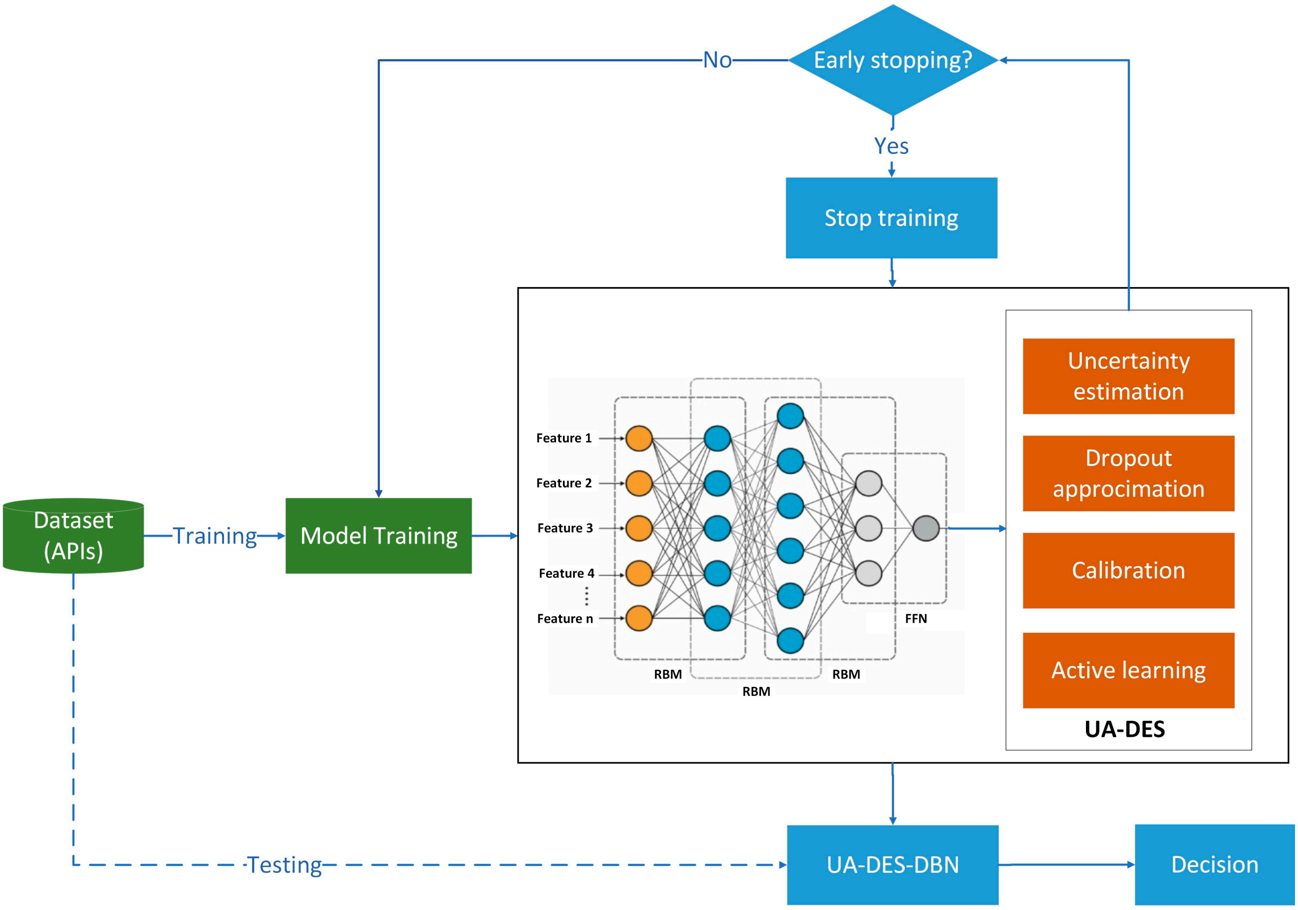

Figure 1. The architecture of the proposed UA-DES training process.

To establish a formal mathematical representation of the UA-DES, we precisely define the essential elements and procedures that are part of the process. This formulation offers a systematic method for implementing UA-DES in training Deep Belief Networks (DBNs) specifically for ransomware detection.

Step 1: Bayesian Performance Modeling

1. Posterior Distribution of Performance Metrics:

Let y be the true label and $\hat{y}$ be the predicted label by the DBN. This label takes two values, ransomware and benign. The performance metric P (e.g., accuracy, precision) can be modeled as:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)}$$

where θ represents the parameters of the DBN, D is the observed data, $P(D \mid \theta)$ is the likelihood of the data given the parameters (model evidence), $P(\theta)$ is the prior distribution of the parameters, and $P(D)$ is the evidence or marginal likelihood of the data. The DBN parameters were set as follows. The batch size is 64, the L2 regulation is 0.0002, the Momentum is 0.7, the dropout rate is 0.2 and the learning rate is 0.05. Those parameters were selected following the standard setup. The model was trained using a set of API-based features selected using the mutual information feature selection.

2. Uncertainty Estimation:

The uncertainty in performance metric P can be quantified using the variance of the posterior distribution:

$$\sigma_P^2 = Var[P(\theta \mid D)]$$

Step 2: Dropout as a Bayesian Approximation

Perform N stochastic forward passes through the DBN with dropout enabled to simulate sampling from the posterior distribution. Dropout is a technique used to prevent overfitting in neural networks by randomly setting neuron activations to zero during training, with a dropout rate hyperparameter set at 0.2. This random deactivation of neurons ensures that the network does not rely on any single neuron and thus learns more robust features. While dropout is consistently applied in training iterations, it is not used during validation or testing, allowing the full network to be evaluated. The training process, with dropout, helps in adjusting the network's weights and biases to develop redundancy in representation, ensuring stability in the network's predictions. The performance metrics from the training phase guide the optimization, while the complete network's capabilities are harnessed during validation and testing for an accurate assessment of its predictive power.

For each pass n, obtain prediction y^n, then calculate the predictive uncertainty as:

$$Uncertainty = \frac{1}{N}\sum_{n-1}^{N} \left(\hat{y}_n - \overline{\hat{y}}\right)^2$$

where $\overline{\hat{y}}$ is the mean prediction over the N passes.

Step 3: Calibration Techniques for Reliable Probability Estimates

After obtaining the raw model output probabilities p, apply a calibration function f to adjust these probabilities:

$$calibrated = f(p)$$

Temperature Scaling (TS), as mentioned above, is used as a calibration technique. This is a post-processing method that adjusts the confidence levels (SoftMax outputs) of a neural network to make them more representative of the true probabilities. The SoftMax function is a mathematical function that transforms a vector of real numbers into a probability distribution by exponentiating each element and normalizing them. The choice of TS was made because it does not change the model's architecture or the relative order of predictions. This calibration method adjusts the model's output probabilities to ensure they accurately represent true threat levels, thus improving the system's ability to differentiate between actual threats and false alarms without causing unnecessary alerts. The integration process involves calibrating the temperature parameter using a validation set post-training, which helps to align the predicted probabilities with actual outcomes, enhancing the model's generalization, reliability, and comprehensibility of its predictions. This makes the UA-DES-DBN model more effective in detecting and managing ransomware threats, ensuring that early stopping does not compromise the predictive quality.

The TS calibration function works as follows. Let z be the logits (i.e., pre-SoftMax activations) output from the DBN model for each class. In other words, the logits are the output values generated by the neural networks before applying the activation function. The standard SoftMax function applied to these logits, which transforms them into probabilities, is given by:

$$P(y = i|x) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

where y is the predicted class, x is the input, z_i is the logit for class i and the denominator sums over all possible classes. Temperature Scaling introduces a temperature parameter T to modify the SoftMax function. The adjusted SoftMax function becomes:

$$P_T(y = i|x) = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$

Here, T is a scalar that adjusts the "sharpness" of the probability distribution. A higher T makes the distribution softer (more uniform), while a lower T makes it sharper. The goal of calibration is to find the optimal temperature T that aligns the model's predictive probabilities with the true likelihood of the outcomes. This is typically performed by minimizing a loss function on a validation set, with the most common choice being the Negative Log-Likelihood (NLL):

$$NLL = \sum_{(x,y)\ni Validation\ Set} \log P_T(y|x)$$

The objective is to find:

$$T^* = \arg \min_T NLL(T)$$

where T^* is the temperature that minimizes the NLL over the validation data.

Once T^* is determined, the model's final output probabilities are adjusted using this optimal temperature:

$$P_{T^*}(y = i|x) = \frac{e^{z_i/T^*}}{\sum_j e^{z_j/T^*}}$$

This calibrated probability distribution is used for all subsequent predictions, ensuring that the model's confidence levels are more aligned with actual outcomes, thus enhancing reliability and decision-making accuracy in the application of the model for ransomware detection.

When referring to "output probabilities" in the context of a Deep Belief Network (DBN) handling ransomware detection, we are discussing the computed probabilities that classify a network event as either ransomware or not ransomware, within a binary classification framework. Specifically, the model generates two probabilities: one that represents the likelihood of an event being ransomware, and another for its being benign, with both probabilities summing to one (i.e., being complementary). This approach, where the network outputs a vector of probabilities for all possible classes—even in binary classification scenarios—is typical in machine learning. This standard practice helps ensure that models are versatile and can be adapted to handle multiple classes, reflecting a general application design prevalent in machine learning environments.

Step 4: Active Learning Framework

Select samples for which the expected information gain, based on the model's current state, is highest. That is, information gain is maximized by selecting and labeling those data samples that yield the highest entropy when evaluated by the model. These high-entropy samples were chosen to maximize the reduction in uncertainty about the model's predictions, thus enhancing its accuracy and robustness through targeted training on the most informative data. By focusing on samples with the highest entropy, the model learns from the most uncertain data points, which generally provide the most informational value. Information gain (IG) for a sample x can be defined as:

$$IG(x) = H(\hat{y}) - E_{p(\theta|D)}[H(\hat{y} \mid x, \theta)]$$

where H is the entropy, representing uncertainty, and $E_{p(\theta|D)}[H(\hat{y} \mid x, \theta)]$ is the expected value over the posterior distribution of the parameters. In our active learning framework, we employ an uncertainty sampling query strategy to efficiently select the most informative data samples for training. This strategy is particularly useful when dealing with large, unlabeled datasets, as it aims to identify and label only those instances where the model's predictions are least confident. The strategy is grounded in measuring the uncertainty of the model's predictions on unlabeled data. We typically use the entropy-based uncertainty measure, defined as follows.

$$H(y|x) = -\sum\nolimits_p P(y = i|x) log P(y = i|x)$$

Here, $H(y \mid x)$ represents the entropy of the predicted probability distribution for a data sample x, where $P(y = i \mid x)$ is the probability that x belongs to class i, as predicted by the model. Higher entropy values indicate higher uncertainty, suggesting that the model is less sure about its prediction.

Step 5: Dynamic Stopping Criteria

Define a stopping criterion based on a combination of accuracy performance improvement threshold ΔP, predictive uncertainty $\sigma^2_{\text{threshold}}$, and calibration quality measure C. The rationale behind selecting these criteria is that early stopping is intended to enhance detection accuracy by reducing the potential for overfitting. Consequently, the aforementioned accuracy-related metrics are the appropriate ones to consider. The effectiveness of this technique was demonstrated in our experimental evaluation, where our model exhibited superior performance compared to those developed in related studies. Stop training if:

$$\Delta P < \epsilon \ \textit{and} \ \sigma_p^2 < \sigma_{threshold}^2 \ \textit{and} \ C > C_{min}$$

where ϵ is a small positive value indicating the minimum acceptable performance improvement, $\sigma^2_{threshold}$ is the maximum acceptable uncertainty level, and C_{min} is the minimum calibration quality threshold. The use of the AND operator in our stopping criteria is primarily intended to optimize protection against overfitting, a prominent concern in ransomware detection. This technique guarantees that the model continues training until it meets all conditions, avoiding early termination based on a single criterion, which could undermine the model's ability to generalize. This is especially important due to the nature of ransomware early detection which should happen as early as before the encryption phase takes place.

The above formalization (i.e., steps 1–5) provides a mathematical foundation for implementing the UA-DES technique, guiding the early stopping decision-making process in a principled manner. By quantitatively assessing performance, uncertainty, calibration, and information gain, this technique aims to optimize the training of DBNs for ransomware detection, thus enhancing model generalization and accuracy. Algorithm 1 shows the pseudocode for the proposed UA-DES and integration into the DBN.

Algorithm 1 Pseudocode describing the proposed UA-DES and integration with the DBN model

```
Initialize DBN model
Initialize training parameters (epochs, learning rate, etc.)
Initialize early stopping parameters (threshold for performance improvement, uncertainty
threshold, calibration quality threshold)
For each epoch in training:
    Train DBN on training dataset
    Evaluate DBN on validation dataset
    Calculate performance metric (e.g., accuracy, F1 score)
    # Dropout as Bayesian approximation for uncertainty estimation
    Perform dropout simulations on validation dataset
    Calculate mean and standard deviation of performance metric across simulations
    # Calibration of probability estimates
    Calibrate model outputs on validation dataset
        Calculate calibration quality (e.g., Expected Calibration Error)
    # Active learning for data efficiency
    If epoch % active_learning_interval == 0:
        Identify and prioritize uncertain samples in training dataset
        Retrain DBN model including prioritized samples
    # Dynamic stopping criterion VALIDATION based on performance improvement,
    uncertainty, and calibration
    If (performance improvement < performance improvement threshold) and
        (standard deviation of performance metric < uncertainty threshold) and\
        (calibration quality > calibration quality threshold):
        stop the training
```

The pseudocode above outlines a structured technique for the integration of our proposed UA-DES into training Deep Belief Networks (DBNs) for ransomware detection, emphasizing the mitigation of overfitting and the enhancement of model generalization. Initially, the algorithm sets up the DBN with specific training parameters and establishes early stopping criteria based on performance improvement, uncertainty thresholds, and calibration quality. Training proceeds in epochs, where the model is iteratively trained and evaluated against a validation dataset to assess its performance. The algorithm utilizes dropout simulations during validation to estimate uncertainty. It calculates the mean and standard deviation of the performance metrics across these simulations, which approximates Bayesian inference. In addition, it adjusts the model's output probabilities to ensure that they accurately represent the actual likelihood of outcomes, as evaluated by metrics like the Expected Calibration Error (ECE). During the training process, an active learning phase is periodically implemented to prioritize samples that provide the most valuable information, thereby improving the efficiency of learning. The algorithm utilizes a dynamic stopping criterion (lines 16 to 19), which stops the training process when there is minimal improvement in performance, low uncertainty, and satisfactory calibration quality. This ensures that the model is neither underfit nor overfit. The process is finalized by saving the trained model, which is then prepared for deployment in tasks related to detecting ransomware. This model combines advanced techniques to create a strong and precise DBN model for detecting ransomware. Precision using the DBN, means that time and again when the circumstances present (i.e., ransomware attacks) the correct classification is achieved (detection).

3.2. The Development of Improved UA-DES DBN for Ransomware Detection

The utilization of the UA-DES mechanism during the training process of a DBN model introduces a novel method for improving the model's precision, specifically in the domain of ransomware detection. The process begins by initializing the DBN with predetermined parameters, such as learning rates and layer configurations. Additionally, early stopping criteria are established, which include thresholds for performance improvement, prediction uncertainty, and calibration quality. During the training process, the model goes through repeated learning cycles called epochs. In each epoch, the model is trained using the training dataset and then assessed using a separate validation set to measure important performance metrics. These metrics evaluate both the model's current ability to detect ransomware and provide a foundation for making dynamic early stopping decisions.

The main advancement of the UA-DES method is its integration of uncertainty estimation using dropout simulations and the calibration of model predictions to guarantee reliability and precision. The algorithm uses dropout simulations to estimate Bayesian posterior distributions, which provide a quantifiable measure of uncertainty in the model's predictions. By calibrating predicted probabilities to accurately reflect true outcomes, this measure offers a more nuanced comprehension of the model's performance and reliability. Active learning enhances the training process by choosing and giving priority to data samples that provide the most useful information, thus increasing the efficiency of the model's learning while simultaneously making it more effective. The dynamic early stopping criterion uses uncertainty and calibration to evaluate performance improvements and ensures that the model stops training at the optimal point. By preventing overfitting, the DBN model becomes robust and accurate in detecting ransomware threats.

3.3. Experimental Environment and Setup

In this section, the experimental results are described. This includes a description of the dataset used and the experimental environment. The experimental study utilized Python packages such as Sklearn, Pandas, Numpy, and SkFeature to analyze a dataset of 8152 crypto-ransomware samples from various families including Cerber, TeslaCrypt, CryptoWall, Petya, and WannaCry. These samples were obtained from the public repository virusshare.com. Additionally, 1000 benign programs were downloaded from informer.com, a well-known repository for Windows applications. Both the ransomware and benign programs were executed in a sandbox environment for the evaluation.

Table 1 displays a sample of API call features that were used as input to the proposed model. These features are directly correlated with the actions typically carried out by ransomware, making them essential for comprehending ransomware behavior. Cryptographic application programming interfaces (APIs) such as CryptEncrypt and CryptGenKey play a crucial role in the process of encrypting files, which is a defining characteristic of ransomware attacks. File access application programming interfaces (APIs), such as CreateFile and DeleteFile, are utilized to access and potentially modify or remove files, indicating unauthorized manipulation of files. Network Application Programming Interfaces (APIs) like WinHttpConnect and WinHttpOpenRequest are crucial for establishing network connections, potentially for the purpose of data exfiltration or command-and-control communication.

Table 1. Feature samples used as input to the proposed model.

Type	Features
Crypto APIs	CryptEncrypt CryptGenKey CryptDestroyKey BCryptGenRandom
File access APIs	CreateFile FindFirstFileEXA FindNextFileA DeleteFile
Network APIs	WinHttpConnect WinHttpOpenRequest

4. Results and Discussion

The accuracy comparison, as shown in Figure 2 between the proposed UA-DES-DBN model and the related models (VGG16-PSO [48], DBN-IDS [49], and DBN [50]) across various input layer sizes reveals that the UA-DES-DBN model consistently outperforms the other models, at almost all input sizes. The results show that as the input layer size increases from 5 to 20, the UA-DES-DBN model's accuracy improves significantly, reaching a peak accuracy of 0.986 at an input size of 20. It maintains a high accuracy level with a slight decrease as the input layer size continues to increase, demonstrating the robustness of our proposed model. The CNN-MD model follows a similar trend but generally exhibits slightly lower accuracy rates than UA-DES-DBN. DBN-IDS and DBN models show more variability and generally lower accuracy rates compared to the proposed model. At an input layer size of 25, all models except the DBN reach their peak accuracy, with UA-DES-DBN achieving 0.982, which is notably higher than DBN-IDS and DBN. Figure 2 illustrates that the proposed UA-DES-DBN model is a strong performer across the board, particularly in the middle range of input layer sizes.

The results obtained from various input layer sizes, as shown in Figure 2, highlight the efficacy of the UA-DES mechanism in improving the performance of the DBN for detecting ransomware. The superior performance of the UA-DES-DBN model, especially with larger input sizes, can be attributed to the UA-DES mechanism's capability to adaptively modify the training process according to the model's uncertainty and calibration quality. This technique enables the model to concentrate on acquiring knowledge from the most informative data points and to stop training before it becomes too specialized, thus maintaining the model's ability to generalize across a wide range of diverse and intricate input patterns. The effectiveness of UA-DES-DBN in utilizing additional information without overfitting is demonstrated by the significant improvement observed, particularly at input sizes of 20 and 40 features. Overfitting, a common issue in deep learning models as they scale, is successfully avoided. The adaptability of the UA-DES is a consequence of its sophisticated method for early stopping, which utilizes Bayesian inference and dropout techniques to estimate uncertainty and make well-informed decisions about when to stop. Hence, the UA-DES not only enhances accuracy rates but also guarantees that these rates are a result of authentic learning and strong model performance. This sets the proposed model apart from similar ones and emphasizes the importance of incorporating advanced early stopping mechanisms in deep learning architectures.

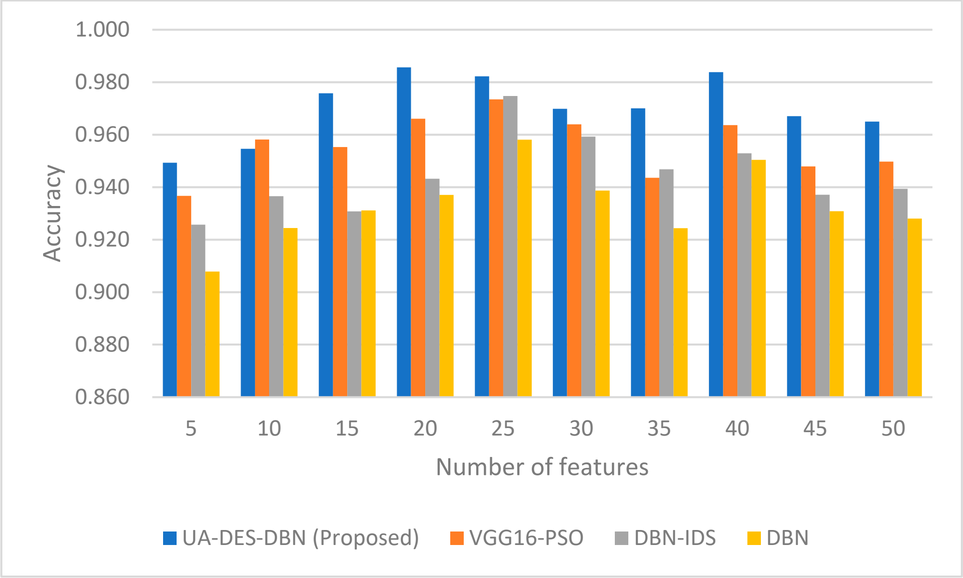

Figure 2. Comparison of accuracy between our proposed UA-DES-DBN and related studies.

From the results in Figure 2, it can be seen that the proposed model maintains better performance than what has been reported in related studies. However, at some points, the difference is rather narrow. The various ransomware evasion techniques, such as polymorphic and metamorphic behaviors, are highly advanced and aim to conceal their presence and imitate harmless software, making detection difficult during the early stages. Our model specifically focuses on the pre-encryption phase of ransomware, which is a crucial juncture in the attack process where the patterns of the attack are scarce and not well-defined. At this stage, ransomware may not display enough unique behaviors to be accurately distinguished from legitimate activities. Therefore, even a slight improvement in the accuracy of detecting ransomware is a noteworthy accomplishment, as it strengthens the model's capability to proactively identify and mitigate the encryption processes before they are fully carried out. Such improvements are crucial for enhancing the model's ability to adapt and thereby accurately identify subtle anomalies that may indicate the beginning of a ransomware attack. This provides a significant advantage in actively defending against these malicious entities including the constant escalation and/or evolving strategies.

The proposed UA-DES-DBN model demonstrates better performance in minimizing the false positive rate compared to related models (CNN-MD, DBN-IDS, and a standard DBN) as shown in Figure 3. This is evident when comparing the false positive rate (FPR) across different input layer sizes. Significantly, as the size of the input layer increases from 20 to 50, the UA-DES-DBN consistently shows lower FPRs compared to other models. This demonstrates its effectiveness in accurately differentiating between harmless and harmful activities without excessively punishing legitimate behavior. At an input layer size of 25, the UA-DES-DBN demonstrates a notably lower FPR of 0.104, highlighting its exceptional ability to effectively process intricate data inputs. This trend demonstrates the model's strength and its capacity to maintain high specificity across various input dimensions. The reduction in FPR, particularly for larger input sizes, can be ascribed to the effective control of the training process by the UA-DES mechanism. This mechanism dynamically adapts based on model uncertainty and calibration quality to avoid overfitting and enhance decision-making thresholds. This technique guarantees that the model is not only adept at identifying ransomware threats but also highly accurate in avoiding false positives, thus improving its practical usefulness in security-conscious environments.

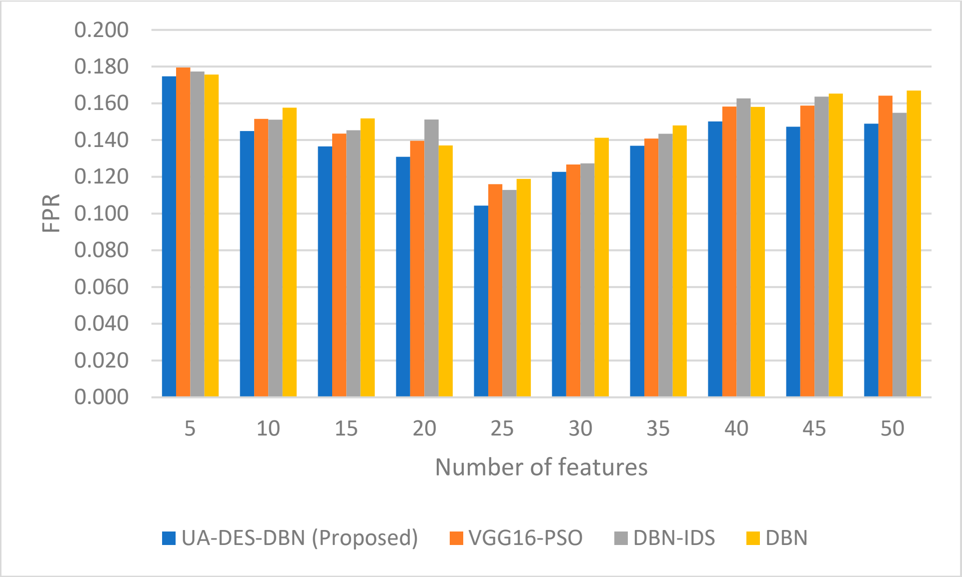

Figure 3. Comparison of FPR between proposed UA-DES-DBN and related studies.

The observed improvements in false positive rate (FPR) shown in Figure 3 measure how the proposed UA-DES-DBN model can be directly linked to the sophisticated early stopping mechanism provided by the UA-DES strategy. This strategy, by incorporating uncertainty estimation and calibration techniques, optimizes the training phase to precisely tune the model's sensitivity to genuine threats while minimizing misclassifications. The gradual reduction in FPR, particularly at higher input sizes (e.g., from 0.104 at size 25 to 0.149 at size 50), exhibits the model's ability to effectively manage the complexity and variability inherent in larger datasets. The UA-DES mechanism enhances the model's discernment capabilities, allowing it to better differentiate between actual threats and benign anomalies, which is critical in ransomware detection (i.e., avoiding false positives). This is achieved by leveraging dropout as a Bayesian approximation to estimate predictive uncertainty and actively adjusting the training process to focus on data points that contribute most significantly to reducing uncertainty. As a result, the model learns a more accurate representation of the threat landscape, leading to lower FPRs. The improvements seen with UA-DES-DBN suggest that the early stopping mechanism not only aids in preventing overfitting but also plays a crucial role in enhancing the model's predictive precision, thus making UA-DES a valuable component in the development of more robust and efficient IDS solutions.

The proposed UA-DES-DBN model is compared to other related models (CNN-MD, DBN-IDS, and a standard DBN) in terms of detection rate (DR), also known as sensitivity, across different input layer sizes. This comparison (as shown in Figure 4) offers valuable insights into their effectiveness in identifying ransomware threats. The UA-DES-DBN model consistently achieves a high detection rate, often outperforming or closely matching the performance of other models, as the input size ranges from 5 to 50. Remarkably, when the input layer size is set to 25, the UA-DES-DBN achieves a DR of 0.956, which is substantially higher than the rates observed in other models. This emphasizes its exceptional capability to accurately identify ransomware attacks. The observed trend indicates that the UA-DES-DBN model effectively utilizes larger input sizes to improve its ability to detect complex and evolving ransomware signatures, which is a crucial advantage. The innovative UA-DES mechanism is responsible for the model's superior detection rates, particularly when dealing with critical input sizes such as 25 and 35. This mechanism enhances the training process by adaptively adjusting to the model's learning progress and uncertainty levels, guaranteeing that the model does not excessively fit the training data and retains its capacity to generalize to novel, unknown ransomware threats. The consistent and high rates of detection, regardless of the size of the input, highlight the robustness of the UA-DES-DBN model. This model has the potential to be a highly effective tool in the field of cybersecurity for combating ransomware.

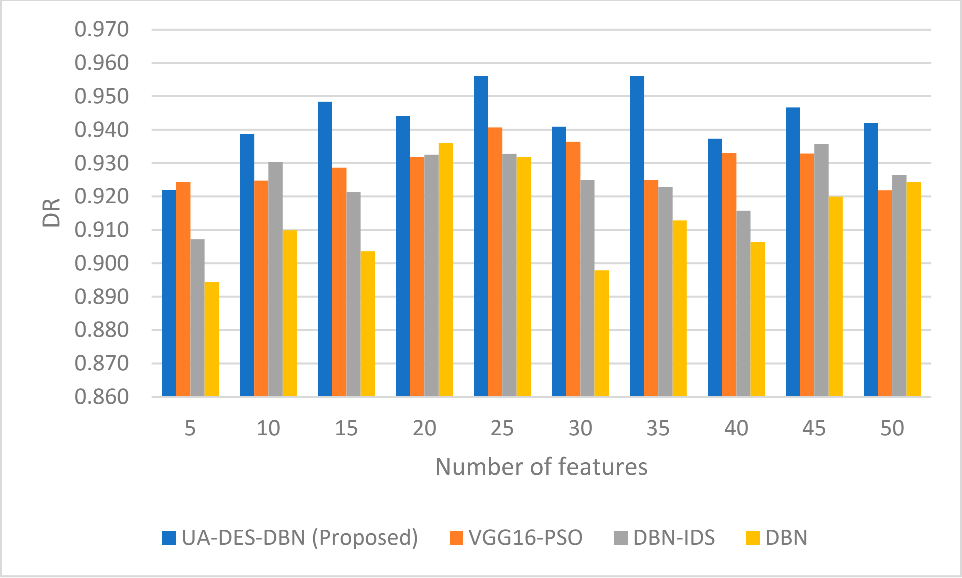

Figure 4. Comparison of DR between proposed UA-DES-DBN and related studies.

The UA-DES-DBN model has demonstrated improved detection rates (in Figure 4) when tested with various input layer sizes. This highlights the effectiveness of incorporating the UA-DES mechanism during the training process. The mechanism's capacity to adaptively modify training time, taking into account real-time evaluations of model performance and uncertainty, is crucial in enhancing the model's sensitivity to ransomware patterns. The UA-DES technique guarantees that the model is not stopped too early or trained excessively, enabling it to effectively extract and learn the most distinguishing features from the input data. The model's performance at medium to large input sizes (such as 25 and 35) can be attributed to UA-DES's ability to deal with the complexity and variability in the data. This allows the model to continuously adapt and enhance its detection capabilities. The UA-DES mechanism utilizes dropout simulations as a substitute for Bayesian inference to achieve a more precise estimation of model uncertainty. This estimation is then used to make an early stopping decision that aims to optimize both high detection rates and low false positives. By following the principles of UA-DES, this customized training method guarantees that the DBN model attains an ideal balance between the depth and breadth of learning, thereby significantly improving its ability to detect ransomware threats. The findings not only emphasize the capability of UA-DES in enhancing the accuracy of detection, but also its contribution in promoting models that are robust, flexible, and extremely efficient in practical cybersecurity scenarios.

The F score comparison between the proposed UA-DES-DBN model and other models (CNN-MD, DBN-IDS, and a standard DBN), considering various input layer sizes, demonstrates the UA-DES-DBN model's ability to achieve a delicate balance between precision and recall in the specific context of ransomware detection (see Figure 5). The proposed model consistently achieves high F scores, demonstrating its capacity to maintain a strong detection accuracy while minimizing both false positives and false negatives. This is a crucial aspect of effective cybersecurity measures. The UA-DES-DBN model demonstrates better capability in accurately classifying ransomware attacks, as evidenced by its F score of 0.962 at an input size of 30. This outperforms the other models and highlights its good performance. The UA-DES-DBN model consistently outperforms its counterparts, particularly in more challenging scenarios characterized by larger input sizes. Although there are slight variations in its F score, the UA-DES-DBN model maintains a competitive advantage across different input sizes. The UA-DES-DBN model consistently achieves high F scores due to the effectiveness of the UA-DES mechanism in optimizing the training process. The UA-DES mechanism ensures that the model's training duration is adjusted dynamically based on its performance and uncertainty levels. This prevents the model from being underfit or overfit, thus improving its ability to generalize effectively to new data. This technique enhances not only the model's ability to detect but also its balance between precision and recall, as indicated by the high F scores. The findings highlight the effectiveness of incorporating advanced early stopping mechanisms such as UA-DES in creating precise and dependable models for detecting ransomware. This further solidifies the UA-DES-DBN model as a valuable asset in the field of cybersecurity.

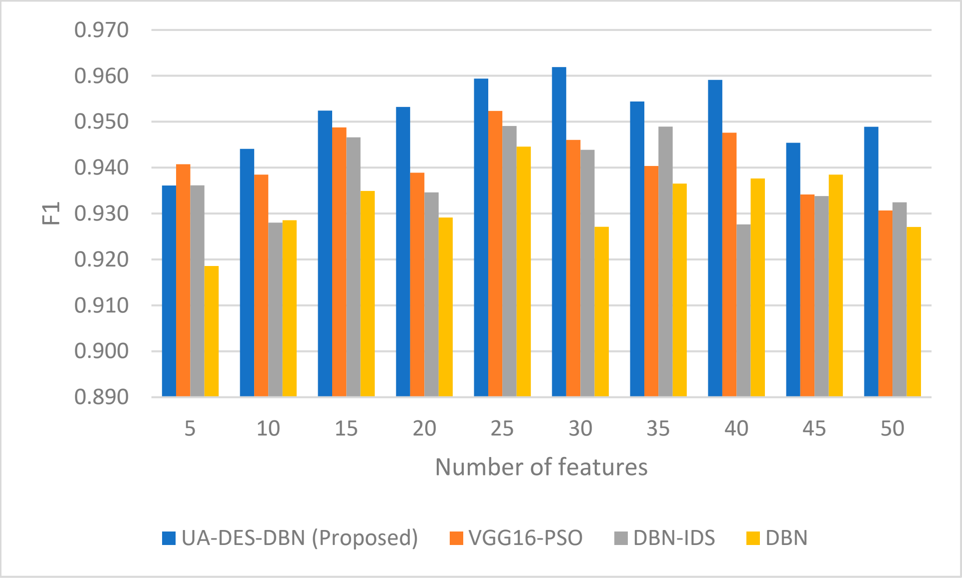

Figure 5. Comparison of F1 score between proposed UA-DES-DBN and related studies.

The F score results (shown in Figure 5) further highlight the suitability of the UA-DES mechanism for improving the performance of the DBN model for detecting ransomware. The increase in F scores for different input sizes, especially when the UA-DES-DBN model performs better than other models, highlights the mechanism's ability to optimize the trade-off between precision and recall, which are crucial components of the F score. The balance between accurately detecting threats and avoiding an excessive number of false alarms is crucial in the field of cybersecurity. The role of the UA-DES is to adjust training cessation criteria based on uncertainty and performance metrics. This ensures that the model accurately captures the complex patterns of ransomware behavior without becoming too specialized to the specific training data. A customized training program enables the model to retain a strong ability to detect attacks while minimizing the occurrence of false positives, as demonstrated by the higher F scores. The UA-DES-DBN consistently performs well regardless of the input size, demonstrating its robustness and adaptability. These attributes are essential in the rapidly changing field of cyber threats. The UA-DES technique utilizes uncertainty estimates to make informed decisions on when to stop training. This not only enhances the model's performance but also improves its efficiency by reducing unnecessary computational loads. The efficiency of incorporating sophisticated early-stopping mechanisms like UA-DES in the development of deep learning models for cybersecurity is demonstrated by its practical benefits and effectiveness. This demonstrates the potential of advanced methodologies to greatly enhance the detection capabilities of models, rendering them indispensable assets in combating ransomware and other cyber threats.

Figure 6 compares the false negative rates (FNR) for the proposed UA-DES-DBN model and other models such as VGG16-PSO, DBN-IDS, and a standard DBN. The analysis is conducted across different input feature sizes, ranging from 5 to 50. The UA-DES-DBN model consistently exhibits lower false negative rates (FNRs), with the rate decreasing as the size of the input layer increases, reaching its minimum at an input size of 25. When the input layer size is reduced to 5, the UA-DES-DBN exhibits a false negative rate (FNR) of 0.150. However, when the input layer size is increased to 50, it consistently maintains a relatively low FNR of 0.128. The observed pattern demonstrates that as the size of the input layer increases, the proposed model shows improved capability in reducing false negatives. However, there are some fluctuations in this trend, as evidenced by an increase in the false negative rate at sizes 30, 35, and 45.

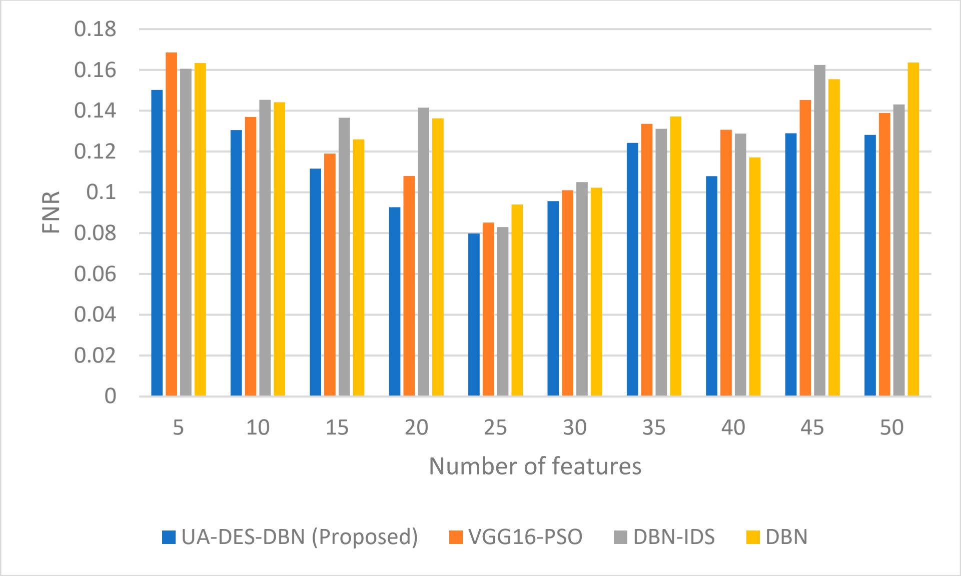

Figure 6. Comparison of False Negative Rate (FNR) between proposed UA-DES-DBN and related studies.

As shown in Figure 6, it is evident that the UA-DES-DBN model excels in reducing false negatives, which are crucial in detecting ransomware. This highlights the model's reliability, as false negatives indicate missed security incidents, making their reduction extremely important. The model demonstrates exceptional performance, especially when the input size is 25. In this case, the False Negative Rate (FNR) is significantly lower compared to the other models. This is attributed to the architecture and training process of UA-DES-DBN effectively capturing the subtle details of ransomware activity within a specific range of input data complexity. The observed rise in false negative rate (FNR) for specific feature sizes indicates a potential overfitting when data dimensionality increases, potentially highlighting the model's susceptibility to changes in the dimensionality of the feature space.

The results in Figure 7, which compared the specificity of the proposed UA-DES-DBN model with VGG16-PSO, DBN-IDS, and standard DBN using input layer sizes ranging from 5 to 50, consistently demonstrate that the UA-DES-DBN model generally outperforms or achieves similar specificity compared to the other models. The UA-DES-DBN model achieves a specificity of 0.896 at an input layer size of 25, which surpasses the performance of all other models. The specificity values of the UA-DES-DBN exhibit a progressive rise until size 25, subsequently displaying minor variations while still maintaining strong performance, thus demonstrating commendable capacity for class differentiation.

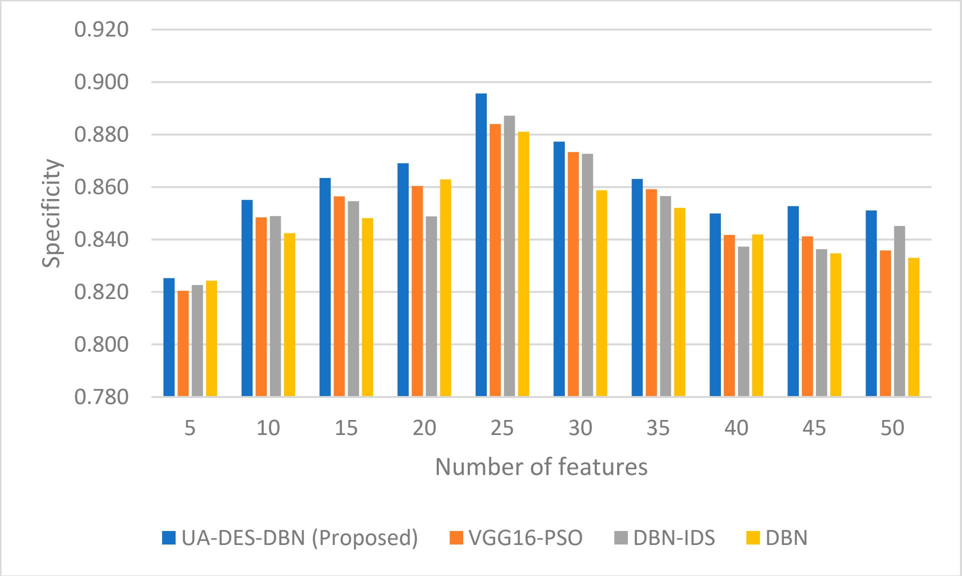

Figure 7. Comparison of specificity between proposed UA-DES-DBN and related studies.

As shown in Figure 7, it is evident that the UA-DES-DBN model consistently achieves higher specificity rates for the various input sizes. This indicates its effectiveness in accurately identifying instances that are not ransomware, which is essential for reducing false positives in ransomware detection. The highest level of performance achieved when the input size is 25 indicates an ideal equilibrium between the complexity of the model and the dimensionality of the input data. This could be a sign of the model's ability to learn efficiently at this particular scale. Nevertheless, the marginal decline in specificity observed beyond this threshold may indicate overfitting or inadequate model adaptability to larger input sizes, necessitating additional investigation and potential model adjustment. The observed trends in specificity support the potential of the proposed model in offering dependable and precise security measures against ransomware. This validates the effectiveness of the UA-DES methodology.

From the results above (Figures 2–7), it can be seen that the proposed model maintains performance levels higher than related studies. However, at some points, the difference was narrow, due to the evasion techniques employed by ransomware (i.e., polymorphic and metamorphic behaviors). As mentioned above, these attacks are highly advanced and aim to conceal their presence and imitate harmless software, making it difficult to detect them in the early stages. Our model specifically focuses on the pre-encryption phase of ransomware, which is a crucial point in the attack process where the patterns of the attack are scarce and not well-defined. At this stage, ransomware may not display enough unique patterns to be accurately distinguished from legitimate activities. Therefore, even a slight improvement in the accuracy of detecting ransomware is a noteworthy accomplishment, as it strengthens the model's capability to proactively identify and mitigate the encryption processes before they are carried out. This improvement is crucial, as it enhances the model's ability to accurately identify subtle anomalies that may indicate the beginning of a ransomware attack. This provides a significant advantage in actively defending against these malicious entities.

Although the use of deep learning algorithms like DBN has gained wide popularity for malware detection (including ransomware), there are some limitations that need further investigation by the research community. One of the potential research directions is to explore the interpretability of the model, which could lead to figuring out where models could fail to detect the attacks. The interpretability makes it easy for researchers to pinpoint the critical aspects of the model that can be improved.

5. Conclusions

The UA-DES technique greatly improves the detection of ransomware in Deep Belief Networks (DBNs) by exploiting the uncertainty and calibration quality measures described herein as was demonstrated through experimentation. This is true because UA-DES enhances the training process using uncertainty estimation and calibration techniques which are different from typical dropout simulations. Another important feature of our methodology is that it represents an active optimization learning framework leading to specific and measurable improvements in overall performance.

During our experiments, the UA-DES-DBN model exhibited an increase in accuracy from 94% to 98% across different input sizes, surpassing other models such as VGG16-PSO by up to 4%, DBN-IDS by up to 6%, and standard DBN by as much as 8%. In addition, the model demonstrates a decrease in the false positive rate from 0.18 to 0.10 (45% savings), indicating a substantial reduction that improves its usefulness in real-world cybersecurity applications. The model's false negative rates also improved, decreasing from 0.15 at an input size of 5 to 0.128 at size 50 (15% savings trend).

These measurements demonstrate our model's robustness to effectively detect ransomware. Future efforts could concentrate on improving the calibration process to enhance the accuracy of probability estimates. The efforts could also explore other deep learning architectures to broaden the model's effectiveness in addressing a wider range of cybersecurity threats using the early stopping criteria defined here. Thus, our ongoing research aims to enhance the detection capabilities against advanced cyber ransomware threats. At this point, our novel ransomware detection methodology has been shown to facilitate the development of a more robust and accurate deep learning methodology to address an ever-changing landscape of advanced and persistent ransomware attacks.

Author Contributions: Conceptualization, M.G. and F.T.S.; methodology, M.G. and F.T.S.; software, M.G.; validation, M.G. and F.T.S.; formal analysis, M.G.; investigation, M.G.; resources, F.T.S.; data curation, M.G.; writing—original draft preparation, M.G.; writing—review and editing, F.T.S.; visualization, M.G.; supervision, F.T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the ransomware portable executable (PE) files are publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jillepalli, A.A.; Sheldon, F.T.; de Leon, D.C.; Haney, M.; Abercrombie, R.K. Security Management of Cyber Physical Control Systems Using NIST SP 800-82r2. In Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 1864–1870.
2. Alqahtani, A.; Sheldon, F.T. Temporal Data Correlation Providing Enhanced Dynamic Crypto-Ransomware Pre-Encryption Boundary Delineation. *Sensors* **2023**, *23*, 4355. [CrossRef] [PubMed]
3. Alqahtani, A.; Gazzan, M.; Sheldon, F.T. A proposed crypto-ransomware early detection (CRED) model using an integrated deep learning and vector space model approach. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 275–279.
4. Gazzan, M.; Sheldon, F.T. An enhanced minimax loss function technique in generative adversarial network for ransomware behavior prediction. *Future Internet* **2023**, *15*, 318. [CrossRef]
5. Gazzan, M.; Alqahtani, A.; Sheldon, F.T. Key factors influencing the rise of current ransomware attacks on industrial control systems. In Proceedings of the 2021 IEEE 11th Annual Computing Workshop and Conference (CCWC), Las Vegas, NV, USA, 27–30 January 2021; pp. 1417–1422.
6. Zakaria, W.Z.A.; Alta, N.M.K.M.; Abdollah, M.F.; Abdollah, O.; Yassin, S.W.M.S. Early Detection of Windows Cryptographic Ransomware Based on Pre-Attack API Calls Features and Machine Learning. *J. Adv. Res. Appl. Sci. Eng. Technol.* **2024**, *39*, 110–131.
7. Alqahtani, A.; Sheldon, F.T. e MIFS: A Normalized Hyperbolic Ransomware Deterrence Model Yielding Greater Accuracy and Overall Performance. *Sensors* **2024**, *24*, 1728. [CrossRef] [PubMed]

8. Gazzan, M.; Sheldon, F.T. Opportunities for early detection and prediction of ransomware attacks against industrial control systems. *Future Internet* **2023**, *15*, 144. [CrossRef]
9. Kapoor, A.; Gupta, A.; Gupta, R.; Tanwar, S.; Detection, I.E.D.R. Avoidance, and Mitigation Scheme: A Review and Future Directions. *Sustainability* **2021**, *14*, 8. [CrossRef]
10. Urooj, U.; Al-Rimy, B.A.S.; Zainal, A.B.; Saeed, F.; Abdelmaboud, A.; Nagmeldin, W. Addressing Behavioral Drift in Ransomware Early Detection Through Weighted Generative Adversarial Networks. *IEEE Access* **2023**, *12*, 3910–3925. [CrossRef]
11. Lee, K.; Lee, S.-Y.; Yim, K. Machine Learning Based File Entropy Analysis for Ransomware Detection in Backup Systems. *IEEE Access* **2019**, *7*, 110205–110215. [CrossRef]
12. Bold, R.; Al-Khateeb, H.; Ersotelos, N. Reducing False Negatives in Ransomware Detection: A Critical Evaluation of Machine Learning Algorithms. *Appl. Sci.* **2022**, *12*, 12941. [CrossRef]
13. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
14. Shaukat, K.; Luo, S.; Varadharajan, V.; Hameed, I.A.; Chen, S.; Liu, D.; Li, J. Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity. *Energies* **2020**, *13*, 2509. [CrossRef]
15. Liu, Y.; Tantithamthavorn, C.; Li, L.; Liu, Y. Deep Learning for Android Malware Defenses: A Systematic Literature Review. *Acm Comput. Surv.* **2022**, *55*, 1–36. [CrossRef]
16. Uysal, D.T.; Yoo, P.D.; Taha, K. Data-Driven Malware Detection for 6G Networks: A Survey From the Perspective of Continuous Learning and Explainability via Visualisation. *IEEE Open J. Veh. Technol.* **2023**, *4*, 61–71. [CrossRef]
17. Urooj, U.; Al-rimy, B.A.S.; Zainal, A.; Ghaleb, F.A.; Rassam, M.A. Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. *Appl. Sci.* **2021**, *12*, 172. [CrossRef]
18. Shemitha, P.A.; Dhas, J.P.M. Crow Search With Adaptive Awareness Probability-Based Deep Belief Network for Detecting Ransomware. *Int. J. Pattern Recognit. Artif. Intell.* **2022**, *36*, 2251010. [CrossRef]
19. Lansky, J.; Ali, S.; Mohammadi, M.; Majeed, M.K.; Karim, S.H.T.; Rashidi, S.; Hosseinzadeh, M.; Rahmani, A.M. Deep Learning-Based Intrusion Detection Systems: A Systematic Review. *IEEE Access* **2021**, *9*, 101574–101599. [CrossRef]
20. Radoglou-Grammatikis, P.; Sarigiannidis, P.; Diamantoulakis, P.; Lagkas, T.; Saoulidis, T.; Fountoukidis, E.; Karagiannidis, G. Strategic Honeypot Deployment in Ultra-Dense Beyond 5G Networks: A Reinforcement Learning Approach. *IEEE Trans. Emerg. Top. Comput.* **2024**, 1–12. [CrossRef]
21. Banaamah, A.M.; Ahmad, I. Intrusion Detection in IoT Using Deep Learning. *Sensors* **2022**, *22*, 8417. [CrossRef] [PubMed]
22. Cao, F. Intrusion Anomaly Detection Based on Pseudo-Count Exploration. *Res. Sq.* **2023**. [CrossRef]
23. Vembu, D.; Ramasamy, G. Optimized Deep Learning-based Intrusion Detection for Wireless Sensor Networks. *Int. J. Commun. Syst.* **2022**, *36*, e5254. [CrossRef]
24. Ferrag, M.A.; Janicke, H.; Smith, R. Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis. In Proceedings of the 6th International Symposium for ICS & SCADA Cyber Security Research 2019 (ICS-CSR), Athens, Greece, 10–12 September 2019. [CrossRef]
25. Cho, H.; Kim, Y.-J.; Lee, E.; Choi, D.; Lee, Y.J.; Rhee, W. Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks. *IEEE Access* **2020**, *8*, 52588–52608. [CrossRef]
26. Dorka, N.; Boedecker, J.; Burgard, W. Adaptively Calibrated Critic Estimates for Deep Reinforcement Learning. *Ieee Robot. Autom. Lett.* **2023**, *8*, 624–631. [CrossRef]
27. Rezaeezade, L.; Batina, A. Regularizers to the Rescue: Fighting Overfitting in DeepLearning-based Side-Channel Analysis. *J. Cryptogr. Eng.* 2022; *under review*. [CrossRef]
28. Moodley, C.; Sephton, B.; Rodríguez-Fajardo, V.; Forbes, A. Deep Learning Early Stopping for Non-Degenerate Ghost Imaging. *Sci. Rep.* **2021**, *11*, 8561. [CrossRef] [PubMed]
29. Kaandorp, M.P.T.; Zijlstra, F.; Federau, C.; While, P.T. Deep Learning Intravoxel Incoherent Motion Modeling: Exploring the Impact of Training Features and Learning Strategies. *Magn. Reson. Med.* **2023**, *90*, 312–328. [CrossRef]
30. Dossa, R.F.J.; Huang, S.Y.; Ontañón, S.; Matsubara, T. An Empirical Investigation of Early Stopping Optimizations in Proximal Policy Optimization. *IEEE Access* **2021**, *9*, 117981–117992. [CrossRef]
31. Choi, H.; Lee, H. Exploiting All Samples in Low-Resource Sentence Classification: Early Stopping and Initialization Parameters. *arXiv* **2021**, arXiv:2111.06971. [CrossRef]
32. Wang, H.; Li, T.H.; Zhang, Z.; Chen, T.; Liang, H.; Sun, J. Early Stopping for Deep Image Prior. *arXiv* **2021**, arXiv:2112.06074. [CrossRef]
33. Li, T.H.; Zhuang, Z.; Liang, H.; Peng, L.; Wang, H.; Sun, J. Self-Validation: Early Stopping for Single-Instance Deep Generative Priors. *arXiv* **2021**, arXiv:2110.12271. [CrossRef]
34. Dai, T.; Feng, Y.; Wu, D.; Chen, B.; Lu, J.; Jiang, Y.; Xia, S.T. DIPDefend: Deep Image Prior Driven Defense against Adversarial Examples. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1404–1412. [CrossRef]
35. Almomani, I.; Qaddoura, R.; Habib, M.; Alsoghyer, S.; Al Khayer, A.; Aljarah, I.; Faris, H. Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data. *IEEE Access* **2021**, *9*, 57674–57691. [CrossRef]
36. Sharmeen, S.; Ahmed, Y.A.; Huda, S.; Koçer, B.Ş.; Hassan, M.M. Avoiding Future Digital Extortion Through Robust Protection Against Ransomware Threats Using Deep Learning Based Adaptive Approaches. *IEEE Access* **2020**, *8*, 24522–24534. [CrossRef]

37. Fernando, D.W.; Komninos, N. A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques. *IoT* **2020**, *1*, 551–604. [CrossRef]
38. Kim, T.; Kang, B.; Rho, M.; Sezer, S.; Im, E.G. A Multimodal Deep Learning Method for Android Malware Detection Using Various Features. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 773–788. [CrossRef]
39. Hemalatha, J.; Roseline, S.A.; Geetha, S.; Kadry, S.; Damaševičius, R. An Efficient DenseNet-Based Deep Learning Model for Malware Detection. *Entropy* **2021**, *23*, 344. [CrossRef] [PubMed]
40. Du, C.; Tong, Y.; Chen, X.; Liu, Y.; Ding, Z.; Xu, H.; Ran, Q.; Zhang, Y.; Meng, L.; Cui, L.; et al. Toward Detecting Malware Based on Process-Aware Behaviors. *Secur. Commun. Netw.* **2023**, *2023*, 6447655. [CrossRef]
41. Fallah, S.; Bidgoly, A.J. Android Malware Detection Using Network Traffic Based on Sequential Deep Learning Models. *Softw. Pract. Exp.* **2022**, *52*, 1987–2004. [CrossRef]
42. Duhayyim, M.A.; Mohamed, H.G.; Alrowais, F.; Al-Wesabi, F.N.; Hilal, A.M.; Motwakel, A. Artificial Algae Optimization With Deep Belief Network Enabled Ransomware Detection in IoT Environment. *Comput. Syst. Sci. Eng.* **2023**, *46*, 1293–1310. [CrossRef]
43. Bharati, S.; Podder, P. Machine and Deep Learning for IoT Security and Privacy: Applications, Challenges, and Future Directions. *Secur. Commun. Netw.* **2022**, *2022*, 8951961. [CrossRef]
44. Ko, E.; Kim, J.-S.; Ban, Y.; Cho, H.; Yi, J.H. ACAMA: Deep Learning-Based Detection and Classification of Android Malware Using API-Based Features. *Secur. Commun. Netw.* **2021**, *2021*, 6330828. [CrossRef]
45. Lu, T.; Du, Y.; Ouyang, L.; Chen, Q.; Wang, X. Android Malware Detection Based on a Hybrid Deep Learning Model. *Secur. Commun. Netw.* **2020**, *2020*, 8863617. [CrossRef]
46. Alghamdi, M.I. Survey on Applications of Deep Learning and Machine Learning Techniques for Cyber Security. *Int. J. Interact. Mob. Technol.* **2020**, *14*, 210–224. [CrossRef]
47. Qiu, J.; Zhang, J.; Luo, W.; Pan, L.; Nepal, S.; Xiang, Y. A Survey of Android Malware Detection with Deep Neural Models. *ACM Comput. Surv.* **2020**, *53*, 126. [CrossRef]
48. El-Ghamry, A.; Darwish, A.; Hassanien, A.E. An optimized CNN-based intrusion detection system for reducing risks in smart farming. *Internet Things* **2023**, *22*, 100709. [CrossRef]
49. Jothi, B.; Pushpalatha, M. WILS-TRS—A novel optimized deep learning based intrusion detection framework for IoT networks. *Pers. Ubiquitous Comput.* **2023**, *27*, 1285–1301. [CrossRef]
50. Sharma, A.; Gupta, B.B.; Singh, A.K.; Saraswat, V. A novel approach for detection of APT malware using multi-dimensional hybrid Bayesian belief network. *Int. J. Inf. Secur.* **2023**, *22*, 119–135. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.