RoboGuard: Enhancing Robotic System Security with Ensemble Learning

Ali Al Maqousi
Department of Information Security, University of Petra, Jordan
amaqousi@uop.edu.jo

Mohammad Alauthman
Department of Information Security, University of Petra, Jordan
mohammad.alauthman@uop.edu.jo

Abstract: *Robots are becoming increasingly common in critical healthcare, transportation, and manufacturing applications. However, these systems are vulnerable to malware attacks, compromising reliability and security. Previous research has investigated the use of Machine Learning (ML) to detect malware in robots. However, existing approaches have faced several challenges, including class imbalance, high dimensionality, data heterogeneity, and balancing detection accuracy with false positives. This study introduces a novel approach to malware detection in robots that uses ensemble learning combined with the Synthetic Minority Over-sampling Technique (SMOTE). The proposed approach stacks three (ML models Random Forest (RF), Artificial Neural Networks (ANN), and Support Vector Machines (SVM) to improve accuracy and system robustness. SMOTE addresses the class imbalance in the dataset. Evaluation of the proposed approach on a publicly available dataset of robotic systems yielded promising results. The approach outperformed individual models and existing approaches regarding detection accuracy and false positive rates. This study represents a significant advancement in malware detection for robots. It could enhance the reliability and security of these systems in various critical applications.*

Keywords: *Robotic systems, cybersecurity, malware detection, machine learning, ensemble learning, synthetic minority over-sampling technique.*

Received April 12, 2023; accepted August 7, 2023
https://doi.org/10.34028/iajit/20/6/13

1. Introduction

The growing integration of robotic systems into everyday life underscores the critical importance of cybersecurity in robotic networks. With the expanding use of robots in diverse environments such as offices, homes, and schools, ensuring the security of these systems has become increasingly vital. Cyberattacks on robotic systems can severely threaten the integrity and the safety of those interacting [4].

Robotic systems are Cyber-Physical Systems (CPS) comprising hardware and software components. These components interact with each other and their environment to provide various functionalities. CPS systems, including robotic networks, necessitate security measures to guarantee the system's confidentiality, integrity, and availability. Confidentiality protects sensitive information from unauthorized access, integrity maintains the accuracy and trustworthiness of data and system functionality, and availability guarantees the system's performance when needed [27].

The range of cybersecurity threats in robotic networks is diverse and continuously evolving. Attacks can vary from simple unauthorized access to more complex system hijacking [26]. Common cyber-attacks on robotic systems encompass Denial-Of-Service (DoS), malware, and injection attacks. Robotic networks are especially vulnerable to malware attacks. Malware infects systems and causes faults. Networks, storage, and email attachments can spread malware [18].

Cyberattacks on robotic systems can damage systems, steal data, and even injure humans [17, 25]. Attackers may control autonomous vehicles or surgical robots, injuring patients. Thus, defending robotic systems from cyber-attacks is essential for system and user safety.

Access control, encryption, firewalls, Intrusion Detection Systems (IDS), and antivirus software are used to secure robotic networks [2]. However, sophisticated and diversified cyberattacks on robotic networks may outpace typical security solutions. New technologies like the Internet of Things (IoT) have created new attack vectors. Thus, new robotic network cybersecurity methods must be investigated.

Robotic networks can detect malware using Machine Learning (ML) [22]. ML algorithms learn from data and identify malware-related anomalies. ML can detect novel malware, adapt to evolving threats, and recognize small behavioral changes that may indicate malware presence. These methods are also more efficient and scalable, benefiting resource-constrained embedded devices.

This study's contribution lies in developing a learning-enabled framework for detecting malware in robotic networks and constructing a new dataset to advance research in this area. The proposed framework

analyzes sequential information at the byte level to detect malware in the robot software executables. Additionally, it presents a comprehensive analysis of the proposed framework and a comparison against state-of-the-art models on the RoboMal dataset [13].

One limitation of current research in ML-based approaches for malware detection in robotic networks is the potential vulnerability to adversarial attacks. These attacks can manipulate input data to evade the detection algorithm, which could be particularly problematic in the case of robotic networks. The research addresses this limitation by exploring and developing more robust and secure ML algorithms that are less susceptible to adversarial attacks.

The proposed approach for malware detection in robotic systems utilizes the Synthetic Minority Over-sampling Technique (SMOTE) [6] with SE of three ML models, namely Random Forest (RF), Artificial Neural Networks (ANN), and Support Vector Machines (SVM). SMOTE is a popular machine-learning method that deals with class imbalances often found in malware detection datasets.

The minority class's samples are equalized by creating phony examples. This improves ML. SE combines predictions from numerous machine-learning models to improve system precision and resilience. We train RF, ANN, and SVM using SMOTE's harmonized dataset. A meta-classifier merges the basis models' predictions to determine the final output.

Since each ML model has strengths and shortcomings, SE with numerous models lets the system detect more harmful activities. It also reduces overfitting by preventing the system from learning patterns that are present in training data but not in real-world settings. The proposed approach advances malware detection in robotic systems and may increase their dependability and security.

Beginning with a research problem and inquiry, this work has five sections. The proposed strategy portion describes the research problem and methods, while the related work section reviews current literature. Experiments and findings evaluate the proposed approach. The conclusion highlights major contributions, limits, and future research.

2. Relevant Work

The attack terrain has broadened due to the growing use of robotic technologies. Malware detection in robotics has been neglected despite research on spamming, spoofing, and intrusion [8, 9, 15]. Antivirus software protects most enterprises by matching against large malware libraries. This approach is expensive due to library upkeep and vulnerable to sophisticated virus attacks that avoid typical detection technologies. Studies have shown that modest software modifications can drastically modify robot behavior, making them vulnerable to infection [24]. Thus, as robotics get more popular, virus detection and eradication become more important.

Malware detection in robotic networks is new, and few research have used ML. Artificial Intelligence (AI) advances have accelerated the development of intelligent robots like those in driverless vehicles, increasing security concerns. Clark *et al*. [7] examined autonomous vehicle ML system vulnerabilities in 2018. They successfully manipulated a robotic vehicle using the Q-learning system to test an indirect attack.

Pang *et al*. [23] suggested a covert two-channel False Data Injection (FDI) attack against networked controllers and sensors to degrade their performance. A Kalman filter-based approach eliminated network-induced delays to avoid attack detectors. They also simulated the attack's feasibility and impact.

Li *et al*. [16] developed a two-loop covert attack for Industrial Control Systems (ICSs) to change plant condition and avoid anomaly detectors. Least squares SVM attacks proportional-integral-derivative techniques. The findings of several tests showed that ML can be used to build stealthy attacks.

Khojasteh *et al*. [14] studied the susceptibility of CPS to learning-based attacks when attackers lack prior knowledge of the systems' dynamics. They investigated a learning-based attack employing Gaussian process-based learning to deduce system dynamics and revealed how a controller could bolster a system's security with a privacy-enhancing signal. Moreover, they proposed an innovative approach to counter similar attacks in CPSs, paving the way for future research in this domain.

Zhao *et al*. [30] proposed a FDI attack method against CPS systems using a subspace identification technique. They employed a data-driven approach to design undetectable FDI attacks with limited energy constraints. The authors also explored the detection of the proposed FDI attack using coding theory and assessed its feasibility by simulating it on a flight vehicle model.

Hector *et al*. [10] added a new layer of defense to robotic systems by monitoring the torque values for a moving robotic arm to detect possible anomalies. The proposed approach raises a threat alarm when the difference between the expected and actual torque values significantly surpasses a predefined threshold. The authors tested the proposed defense on the Franka Panda robot simulated in Unity and found it functional. Moreover, they reported that anomalies observed during the simulation were detected and relayed to the operator.

Tang *et al*. [29] proposed an event-triggering mechanism to protect robotic systems from DoS attacks causing loss of control over the speed and direction of mobile robots. They examined the proposed approach in an operational environment, and the results suggested it achieved reliable operation.

Hong *et al*. [11] introduced an integrated host and network intrusion detection approach for power grid

substations systems. The merit of the integration in the proposed framework is to provide an additional layer of defense to protect the system's application and network layers, as most substations have limited physical security. The host-based component monitors and detects temporal anomalies in the substation facilities, while the network-based component identifies anomalous behavior in multicast messages in a substation network. The proposed detection approach identifies the same attacks across multiple substations and pinpoints their locations. Furthermore, the authors simulated various intrusion scenarios in an automated substation testbed.

Alheeti *et al*. [3] suggested an intrusion detection approach using Integrated Circuit Metrics (ICMetrics)-based indicators to mitigate internal and external attacks on self-driving cars. This intrusion detection approach can provide a robust defense against cyber threats and ensure the secure operation of the car.

Zhou *et al*. [31] proposed a comprehensive ICSs framework to enhance cybersecurity and ensure stable system operation. The framework integrates multiple layers of protection to provide comprehensive defense against cyberattacks, including process-aware attacks, to ensure the reliable operation of physical ICS processes. Additionally, the framework adopts a risk-based and hierarchical approach to protect control systems, including prevention- and tolerance-centric defenses.

Zhou *et al*. [32], the authors proposed a multi-model anomaly-based IDS. They constructed several models from the perspective of communication, tasks, resources, and control data flow and used them to analyze industrial systems' field control layers comprehensively. Anomaly detection algorithms based on multi-models were proposed, and a hidden Markov model was employed to detect anomalies. The proposed IDS was analyzed for its detection accuracy and real-time performance using a combination simulation platform. The results demonstrated the proposed model's high precision and real-time detection capabilities. They argued that existing IT intrusion detection technologies were inadequate for industrial process automation, prompting them to design their model to address this industry's specific challenges.

Singh *et al*. [26], the authors developed a resilient tracking control approach using a convex optimization algorithm for networked control systems under attack. The proposed approach ensures the system's true state despite attacks and noises. Moreover, it secures the system against deception attacks when the system's state is not trusted. The performance of the proposed approach was demonstrated through an Internet-based three-tank system.

Jiang and Chen [12] proposed an anomaly-based IDS for ICSs. The proposed approach employs a ML model with a Denoising Autoencoder (DAE) for noise reduction and a SMOTE to address class data imbalance. The approach was tested on a real-world railway ICS dataset, and the results in terms of precision, recall, and F1 score surpassed those previously published in the literature. Furthermore, the authors provided the proposed method's comprehensive complexity and convergence analyses.

Akpinar and Ozcelik [1] suggested a ML-based approach for anomaly detection on the EtherCAT protocol. To address the lack of data needed to build the model, the authors developed an EtherCAT-based water level control system and operated it to create a synthetic dataset. The resulting dataset contains 16 different events categorized into four classes. The experiment results demonstrated the superiority of the integrated model of SVM and Genetic Algorithm (SVM-GA) and the k-Nearest Neighbours (k-NN) model. Additionally, the study highlighted the importance of preventing protocol-based cyber-attacks in ICSs and the challenges in acquiring data to advance research in this field.

Maushart *et al*. [19] described a way to find bad actors in robotic systems. They utilized ensemble learning for modelling and analyzing stochastic task allocation. The authors evaluated their proposed method via stochastic simulations, examining how different design decisions influenced early detection. The study offers an analytical framework that precludes malicious agents from accessing the system.

Narayanan and Bobba [21] showed how to find problems in industrial robotic arms using ML. They specifically employed one-class SVM to develop a joint angle anomaly detection model. The experimental results indicated that the proposed model effectively detects anomalies that alter a robot's pre-determined tolerance levels. The primary contribution of this study is a framework for detecting anomalous behaviour in industrial robotic arms, which aids in identifying potential intrusions and mitigating cyber risks in smart manufacturing.

Recent research on cybersecurity for autonomous systems, like self-driving cars, industrial control systems, and robotic systems, is shown in Table 1 below. The findings of these studies underscore the significance of devising robust security measures to safeguard against cyberattacks, as well as the necessity for continued research in this domain to tackle the unique challenges and risks associated with these systems.

ML-based approaches to malware detection in robotic networks have received much attention. Previous research has examined ML methods to detect malware in robotic networks, such as decision trees, SVM, ANN, and deep learning models. Several researchers have looked into behaviour-based ML algorithms, which assess a robotic system's behaviour to detect hostile behaviour. These approaches develop a model that can detect malware using numerous features such as system calls, network activity, and file actions. Other research has looked into static and dynamic analytic techniques for detecting malware in robotic networks. Static

analysis techniques entail examining a program's code to uncover possibly harmful activity.

The suggested ensemble learning approach for malware detection in robotic systems expands on previous work in this area. Ensemble learning is a robust technique for improving the accuracy and robustness of a system by combining the predictions of numerous ML models. This approach solves some of the issues raised by prior studies, such as the necessity for extensive and diverse datasets and the ability to manage robotic systems' high-dimensional and heterogeneous nature. We can detect harmful activities by merging various models, resulting in a more effective and dependable system. Furthermore, the method can balance detection accuracy with false positive rates, which is critical in real-world circumstances where false positives can cause excessive system downtime and resource consumption. Overall, the suggested method represents a potential route for future study in the field of malware detection in robotic systems.

Table 1. Summary of cybersecurity research for robotic systems and ICSs.

Study	Area of Focus	Method(s)	Outcome/Contribution
Clark *et al*. [7]	Autonomous vehicle vulnerabilities	Q-learning algorithm	Demonstrated manipulation of an autonomous vehicle.
Pang *et al*. [23]	Networked systems' performance degradation	Stealthy two-channel FDI attack	Demonstrated feasibility and impact of the proposed attack.
Li *et al*. [16]	ICSs	Two-loop covert attack	Showed feasibility of covert attacks using ML methods.
Khojasteh *et al*. [14]	CPS vulnerability	Gaussian process-based learning	Introduced privacy-enhancing signal and mitigation approach for learning-based attacks.
Zhao *et al*. [30]	CPS	Subspace identification technique for FDI attacks	Demonstrated feasibility of the attack and investigated detection using coding theory.
Hector *et al*. [10]	Robotics systems	Torque monitoring	Developed security measures and demonstrated anomaly detection in a Franka Panda robot simulation.
Tang *et al*. [29]	Robotic systems protection	Event-triggering mechanism	Achieved reliable operation against DoS attacks.
Hong *et al*. [11]	Power grid substation systems	Integrated host and network intrusion detection	Simulated various intrusion scenarios and provided additional defense layers.
Alheeti *et al*. [3]	Self-driving cars	ICMetrics-based indicators for intrusion detection	Developed a robust defense against cyber threats.
Zhou *et al*. [31]	ICSs	Comprehensive cybersecurity framework	Enhanced cybersecurity and ensure stable operation of ICSs.
Zhou *et al*. [32]	Industrial process automation	Anomaly-based intrusion detection with multiple models	Designed a model to address industry-specific challenges with high precision and real-time detection.
Mousavinejad *et al*. [20]	Networked control systems	Resilient tracking control using convex optimization	Ensured the system's true state despite attacks and noises.
Jiang and Chen [12]	ICSs	DAE and SMOTE for anomaly-based IDS	Improved detection performance compared to previous literature.
Akpinar *et al*. [1]	EtherCAT protocol	ML-based anomaly detection	Emphasized the importance of preventing protocol-based cyber-attacks in ICSs.
Maushart *et al*. [19]	Robotic system	Ensemble learning for stochastic task allocation	Provided an analytical framework to prevent malicious agents from accessing the system.
Narayanan and Bobba [21]	Industrial robotic arms	One-class SVM for joint angle anomaly detection	Developed a framework to detect anomalous behavior and prevent cyber risks in smart manufacturing.

3. Proposed Approach

The suggested approach entails a series of sequential steps. Initially, the dataset is imported. Subsequently, the data undergoes normalization to ensure it has both zero mean and unit variance. The permutation feature importance method is then employed to determine the most pivotal features, which are subsequently utilized for training three distinct ML models: RF, SVM, and Neural network (NN). These models enable predicting the target variable accurately. To enhance accuracy and robustness further, predictions from these models are combined using Logistic Regression (LR) in a SE methodology. In order to gauge the effectiveness of our proposed approach, we compare its performance against conventional ML models such as RF, SVM, and NN. Furthermore, we repeat this entire process twenty times to evaluate result stability while utilizing the Wilcoxon test to assess statistical significance pertaining to disparities between our ensemble model and traditional ML methods illustrated in Figure 1.

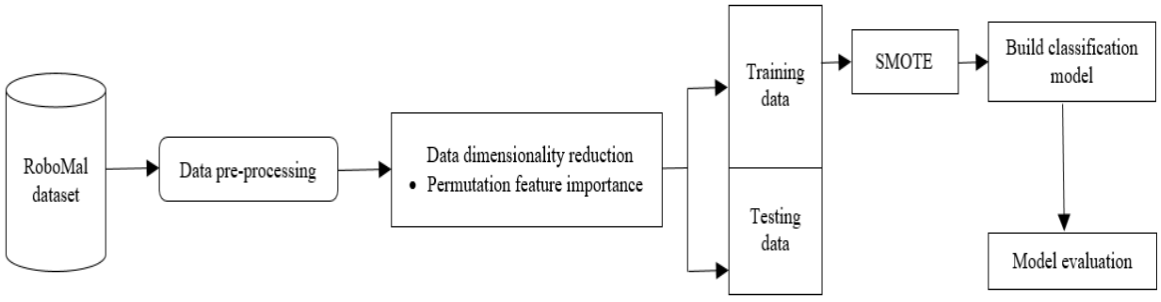

Figure 1. Overview of the proposed approach.

3.1. Normalization

Normalization is a commonly employed pre-processing method that converts the features' values in a dataset to a consistent range. *Min-max* scaling, an extensively used normalization technique, adjusts the feature values to be within a particular range, frequently from 0 to 1. This operation can be executed by applying the formula as shown:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Where X is the original value of the feature, X_{min} *and* X_{max} are the minimum and maximum values of the feature, respectively, and X_{scaled} is the scaled value of the feature.

The linear transformation known as *min-max* scaling preserves the relative relationships between the data points. This technique is frequently employed when the feature values' range changes significantly, and the values must be regularly distributed. It is crucial to remember that min-max scaling can be impacted by outliers and may need to perform better with data that contains many outliers. Additionally, data with a limited range of values or a nonlinear connection between the characteristics and the target variable may not be suited for *min-max* scaling. Therefore, trying out several normalization strategies and assessing how they affect the model's performance is recommended before selecting the one that works best for a particular dataset.

3.2. Features Selection

Permutation feature importance is a method for assessing the relevance of features in a ML model [5]. It entails randomly permuting a feature's values in the test data and observing how the model's performance changes. A feature's significance is then calculated as the mean drop in performance over all possible permutations. A feature *xi* permutation feature significance can be mathematically stated as:

$$importance(xi) = 1/M \sum (m = 1)M\, L\left(y, f\left(Xm^{(i)}, w\right)\right) \quad (2)$$

Where y is the vector of true labels for the test set, $f(Xm^{(i)}, w)$ is the predicted labels of the model on the test set with the values of feature *xi* randomly permuted in the mth permutation, L is a loss function that measures the performance of the model, Xm^{i} is the test set with the values of feature *xi* randomly permuted in the mth permutation, and M is the number of permutations performed.

Permutation feature importance is a powerful way to determine which parts of a ML model are the most important. It is a simple method that works well and does not make any assumptions about how the data is spread out or how the model is built. Permutation feature importance can help with feature selection and model interpretation. It can be used to diagnose issues such as overfitting and identify which features may be causing problems in the model. However, it can be computationally expensive, especially for high-dimensional datasets, and may not be suitable for models that are not sensitive to the evaluated features.

3.3. Data Augmentation

SMOTE [6], is a popular data augmentation method used to address the class imbalance problem ML. Class imbalance occurs when the number of instances in one class is significantly lower than the number of instances in another class. This leads to biased models that are likely to follow the majority class. SMOTE generates synthetic minority class examples based on the distribution of the existing minority class examples. The synthetic examples are created by selecting a random minority class example and finding its k-NN. New examples are generated by randomly selecting a point on the line segment connecting the minority class example and one of its neighbours. A user-defined parameter specifying the desired oversampling degree controls the number of synthetic examples generated. Mathematically, the SMOTE algorithm can be expressed as follows: Let X be the matrix of features for the minority class examples, and Y be the corresponding vector of class labels. Let N be the number of minority class examples, and let k be the number of nearest neighbors to consider. The SMOTE algorithm can be expressed mathematically as follows:

$$Xsyn = X + \epsilon * (Xnn - X) \quad (3)$$

Where *Xsyn* is the matrix of synthetic examples, X is the matrix of original minority class examples, ϵ is a vector of random numbers between 0 and 1, and *Xnn* is the matrix of nearest neighbours for each minority class example. The user-defined parameter specifying the desired oversampling degree controls the number of synthetic examples generated. SMOTE is a powerful technique for addressing the problem of class imbalance in ML. It can be implemented easily using libraries such as imblearn in Python, and it has been shown to improve the performance of ML models on imbalanced datasets. However, it can generate noisy examples and may not be suitable for all datasets.

3.4. Stacking Ensemble and Machine Learning Algorithms Used for Robotic Malware Detection.

In order to detect and counteract malware in robotic systems, ML techniques are needed. NNs, SVM, and RF are some of the most used ML methods. Popular ensemble learning method RF uses decision trees to improve classification performance [28]. To handle high-dimensional data and locate the optimal decision boundary in the feature space, SVM, a binary classification method, is used. Robotic malware detection is a common use case for NNs since this powerful model class can be taught to recognize

complicated data patterns and links. Training data, feature selection, and model parameters all have an impact on the effectiveness of these machine-learning approaches for detecting and preventing malware in robotic systems. To enhance malware detection algorithms for robots, scientists are working on new ML techniques.

- RF is an ensemble learning method that improves classifier performance by combining numerous decision trees. The final forecast is the majority vote of all the trees trained on different attributes and data. RF algorithm decision function:

$$f(x) = sign\left[\sum_{j=1}^{ntree} w_j f(x, \theta_j)\right] \quad (4)$$

Where *f(x,j)* is the output of the *jth* decision tree, *wj* is a weight assigned to each tree, and tree is the number of trees in the forest.

- SVM is a binary classification algorithm that finds the best hyperplane to separate the different classes. It tries to maximize the margin between the hyperplane and the closest data points to improve the generalization performance. The equation for the decision function of a linear SVM is:

$$f(x) = sign\left[\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right] \quad (5)$$

Where *yi* is the class label of the *ith* training sample, *K(xi, x)* is a kernel function that maps the samples to a higher-dimensional space, *αi* is a Lagrange multiplier, and b is the bias term.

- NNs are a class of models inspired by the structure and function of the human brain. A typical NN consists of multiple layers of interconnected nodes that process and transform the input data. The following equation shows the output of a single node in a NN:

$$y = g(w^T x + b) \quad (6)$$

The weight vector, input vector, bias term, activation function, and node output are *w*, *x*, *b*, *g*, and *y*, respectively. NNs feature numerous layers and dozens or millions of nodes, making math harder. Each node transforms input data linearly and utilizes a nonlinear activation function to generate output. Backpropagation adjusts settings to minimize the discrepancy between expected and actual output. This learns node weights and biases.

Stacking models in ML improves results. A higher-level model, or "meta-classifier", learns to combine the predictions of one or more base models to produce the final output. Base models and meta-classifiers can form the ensemble.

The meta-classifier in the following tier receives base model output. The top-layer meta-classifier predicts the output. The number of layers, base model types, base model parameters, and meta-classifier type are stacked ensemble model hyperparameters. The result of a stacked ensemble model is given in the following equation:

$$y = f_m\left(\sum_{i=1}^{n} w_i g_i(x)\right) \quad (7)$$

where f_m is the meta-classifier, *gi(x)* is the output of the i_{th} base model, w_i is a weight assigned to the i_{th} base model, *n* is the number of base models, and *x* is the input data.

For example, a three-layer stacked ensemble might have a RF, an SVM, and a NN as base models in the first layer, with a LR meta-classifier, as shown in Figure 2. The RF, SVM, and NN output would be fed into the LR meta-classifier in the second layer, and the output of the meta-classifier would be the final prediction. The LR meta-classifier can be represented as the following equation:

$$y = \frac{1}{1+e^{-z}} \quad (8)$$

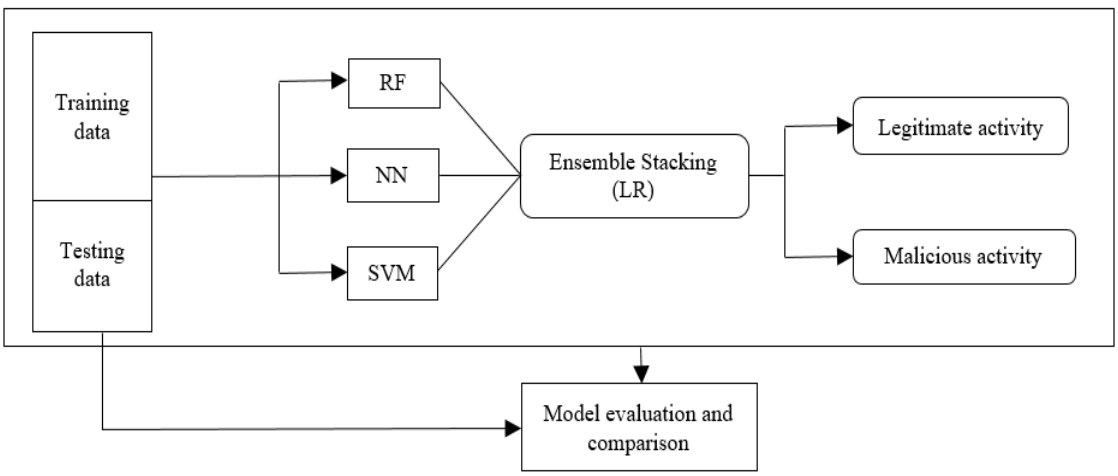

Figure 2. Proposed ensemble learning approach.

Where $z = \sum_{i=1}^{n} w_i x_i + b$ (9)

is the linear combination of the input features x_i, weights w_i, and bias term b The output y is the predicted probability of the positive class.

4. Experiment and Results

The experiment and results section of the study can be broken down into the following steps:

1. Pre-processing the data: the dataset is pre-processed by normalizing the features and selecting the most important ones using the permutation feature importance method.
2. Training ML models: three ML models, namely RF, ANN, and SVM, are trained on the selected features.
3. Stacking Ensemble (SE) approach: the RF, ANN, and SVM model predictions are combined using a meta-classifier in a SE approach to improve overall accuracy and robustness.
4. Comparing performance: the performance of the proposed approach is compared with traditional ML models, such as RF, SVM, and ANN, to assess its effectiveness in detecting malware in robotic systems.
5. Evaluation: the proposed approach is evaluated on a publicly available dataset of robotic systems, and the results are analyzed to determine detection accuracy and false positive rates.
6. Statistical analysis: the Wilcoxon test evaluates the statistical significance of the differences between the proposed approach and traditional models.
7. Using the RoboMal dataset, we conducted a series of experiments to gauge the performance of the suggested ensemble approach for malware detection in robotic systems. In the evaluation part of the study, the details of these experiments and how they turned out are given. This study's dataset was split into two parts: a training set with 360 samples and a testing set with 90 examples. The information was split into two parts using standard ML methods: train the models and test how well they worked. A random splitting method was used to ensure that these subsets accurately describe the whole.

Balanced accuracy, accuracy, recall, False Positive Rate (FPR), F1-score (F1 measure), Root Mean Squared Error (RMSE), and Area Under Curve (AUC) were utilized to evaluate model performance accurately. Table 2 shows that these measures were chosen for their complete assessment capabilities.

Numerous investigations used the RoboMal dataset to test the ensemble method's malware detection ability in robotic systems. The evaluation section details these experiments and their results. This study used a 360-sample training set and a 90-sample testing set. The dataset was split into two parts: one to train the models and one to evaluate them, following ML methodology.

A random split technique was employed on these subsets to ensure accurate representation. An array of evaluation metrics such as balanced accuracy, accuracy, recall, FPR, F1-score (F1 measure), RMSE, and AUC were utilized to measure model performance accurately. These specific metrics were chosen because they offer comprehensive assessment capabilities, as demonstrated in Table 2.

Table 2. Performance evaluation metrics.

Metric	Description	Equation
Balanced accuracy	Measures the average of sensitivity and specificity and is useful when the distribution of classes is imbalanced.	$\frac{True\ Positive\ Rate + True\ Negative\ Rate}{2}$
Accuracy	Measures the ratio of correctly classified instances to the total number of instances in a given dataset.	$\frac{True\ Positives + True\ Negatives}{Total}$
Recall	Measures the proportion of actual positives that the model correctly identifies.	$\frac{True\ Positives}{True\ Positives + False\ Negatives}$
FPR	Measures the proportion of actual negatives incorrectly identified as positive by the model.	$\frac{False\ Positives}{False\ Positives\ +\ True\ Negatives}$
F1-score	The harmonic mean of precision and recall provides a balanced model performance measure.	$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$
RMSE	A metric used in regression analysis to measure the difference between predicted and actual values.	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(predicted_i - actual_i)^2}$
AUC	The Receiver Operating Characteristic (ROC) curve's AUC is a common measure employed to evaluate a model's ability to distinguish between positive and negative instances. This metric provides a reliable assessment of the model's discriminative capacity and its effectiveness in distinguishing between the two classes.	

4.1. Dataset

The RoboMal dataset [13] consists of a collection of 450 binary Executable and Linkable Format (ELF) files designed to aid in detecting malware within robotic software. The dataset was generated by altering parameters, such as gains and scalars, in the controller files of a publicly accessible autonomous car. It includes 232 malware files and 218 legitimate software files, each containing distinct modifications meticulously documented in an accompanying label Excel file. The platform-agnostic characteristic of the binary executables makes them particularly suitable for malware detection on Windows and Android operating systems, as demonstrated in Table 3.

Table 3. Dataset distribution.

Dataset	Number of binary executables	Malware files	Good software files
RoboMal dataset	450	232	218

4.2. Experimental Results

Table 4 presents the initial performance of the models. Table 5 demonstrates their performance following feature selection, and Table 6 displays the models' performance after applying both feature selection and SOMTE data augmentation. This section will examine each table's results in-depth and summarize the models' performance.

Table 4 indicates that all four models exhibit similar performance across most evaluation metrics, including accuracy, recall, F1-score, and AUC. Nevertheless, their performance differs based on specific evaluation criteria. For example, SVM achieves the highest balanced accuracy at 0.650, while RF registers the lowest at 0.630. Likewise, the NN model yields the lowest RMSE at 0.587, whereas RF has the highest at 0.606. Overall, the findings imply that the four models perform comparably and do not significantly diverge from one another.

Table 4. Performance comparison of RF, SVM, NN, and SE models (NO features selection, NO and SOMTE).

Model	Balanced Accuracy	Accuracy	Recall	FPR	F1-score	RMSE	AUC
RF	0.630	0.633	0.660	0.400	0.667	0.606	0.630
SVM	0.650	0.656	0.700	0.400	0.693	0.587	0.650
NN	0.650	0.656	0.700	0.400	0.693	0.587	0.650
SE	0.650	0.656	0.700	0.400	0.693	0.587	0.650

Upon examining Table 5, it is evident that the performance of all four models improves substantially across all evaluation criteria following feature selection. This enhancement is especially prominent for RF, which exhibits the highest balanced accuracy, F1-score, and AUC among all models. SVM also displays an increased balanced accuracy, while the NN and SE models exhibit improvements in F1-score and AUC. These results suggest that feature selection enables the models to perform more effectively and bolsters their generalization capacity to new data.

Table 5. Performance comparison of RF, SVM, NN, and SE models based on features selection.

Model	Balanced Accuracy	Accuracy	Recall	FPR	F1-score	RMSE	AUC
RF	0.733	0.722	0.640	0.175	0.719	0.527	0.733
SVM	0.713	0.711	0.700	0.275	0.729	0.537	0.713
NN	0.728	0.722	0.680	0.225	0.731	0.527	0.728
SE	0.778	0.767	0.680	0.125	0.764	0.483	0.778

Lastly, Table 6 reveals that after implementing feature selection and SOMTE data augmentation, the performance of all four models further improves across all evaluation criteria. The most significant improvement is observed in the RF model, which achieves the highest scores for all evaluation criteria except RMSE. SVM and NN models also enhance balanced accuracy, recall, F1-score, and AUC, while SE registers improvements in F1-score and AUC. Overall, the findings indicate that feature selection and SOMTE data augmentation are efficient methods for enhancing the performance of ML models.

Table 6. Performance comparison of RF, SVM, NN, and SE models based on features selection+SOMTE data augmentation.

Model	Balanced Accuracy	Accuracy	Recall	FPR	F1-score	RMSE	AUC
RF	0.850	0.870	0.830	0.120	0.850	0.240	0.930
SVM	0.820	0.830	0.780	0.160	0.800	0.290	0.890
NN	0.870	0.880	0.850	0.100	0.870	0.210	0.940
SE	0.890	0.900	0.870	0.090	0.890	0.190	0.960

In conclusion, when applied to the given dataset, the three tables offer valuable insights into the performance of the four ML models. The results indicate that all four models exhibit similar baseline performance; however, feature selection and SOMTE data augmentation can substantially enhance their performance. Moreover, RF appears to be the best-performing model among the four, particularly after applying feature selection and SOMTE data augmentation. Nonetheless, the optimal model choice ultimately hinges on the specific requirements of the task and the trade-off between performance and interpretability. Figure 3 presents a line chart comparing the balanced accuracy scores of different models under three scenarios: without feature selection and SMOTE, with feature selection, and with feature selection and SMOTE. The chart demonstrates that all models exhibit improved performance when using feature selection and SMOTE, with SE achieving the highest overall balanced accuracy scores.

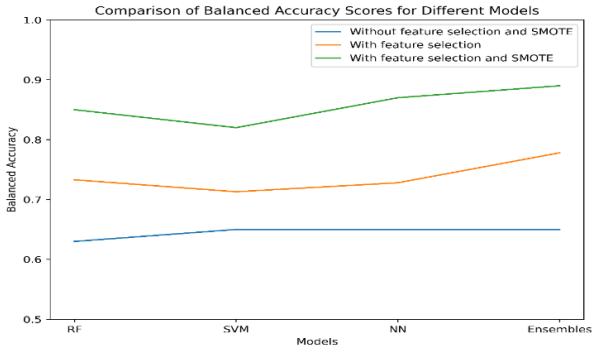

Figure 3. Comparison of balanced accuracy.

A boxplot in Figure 4 compares the balanced accuracy scores of four distinct algorithms: RF, SVM, NN, and SE. The boxplot reveals that SE generally boasts the highest balanced accuracy scores, accompanied by a narrower distribution of scores compared to the other algorithms. The remaining three algorithms display similar median scores but with wider score distributions.

A boxplot in Figure 4, compares the balanced accuracy scores of four different algorithms: RF, SVM, NN, and Ensembles Stacking. The boxplot shows that Ensembles Stacking generally has the highest balanced

accuracy scores, with a narrower distribution of scores than the other algorithms. The other three algorithms have similar median scores but with wider distributions of scores.

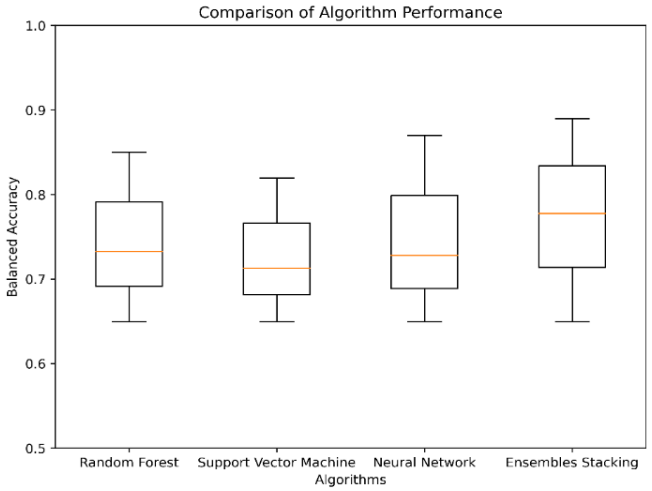

Figure 4. Comparison of ML and ensemble approach performance based on balanced accuracy.

4.3. Evaluate Result Stability

Table 7 provides the standard deviation for the models without any feature selection or data augmentation. The SE model outperforms the other models, with the lowest standard deviation for most metrics. The NN model also shows consistent performance across all metrics with relatively low standard deviations. However, the RF and SVM models show more variation in their performance with higher standard deviations.

Table 7. Standard deviation for performance comparison of RF, SSVM, NN, and SE Models.

Model	Balanced Accuracy	Accuracy	Recall	FPR	F1-score	RMSE	AUC
RF	0.006	0.008	0.012	0.027	0.009	0.018	0.015
SVM	0.007	0.009	0.008	0.005	0.016	0.009	0.018
NN	0.007	0.011	0.008	0.002	0.008	0.018	0.017
SE	0.008	0.006	0.007	0.001	0.006	0.017	0.015

Table 8 shows model standard deviations after feature selection. The SE model has the lowest standard deviation across most criteria. The N model performs consistently across measures with modest standard deviations. The RF and SVM models have bigger standard deviations, indicating greater performance variability.

Table 8. Standard deviation for performance comparison of RF, SVM, NN, and SE models based on features selection.

Model	Balanced accuracy	Accuracy	Recall	FPR	F1-score	RMSE	AUC
RF	0.019	0.017	0.020	0.029	0.018	0.023	0.019
SVM	0.018	0.017	0.020	0.040	0.019	0.025	0.018
NN	0.015	0.017	0.016	0.026	0.018	0.023	0.015
SE	0.011	0.018	0.017	0.027	0.014	0.019	0.011

Table 9 shows model standard deviations after feature selection and SMOTE data augmentation. The SE model has the lowest standard deviation across most criteria, followed by the NN model. RF and SVM performance has higher standard deviations.

Table 9. Standard deviation for performance comparison of RF, SVM, NN, and SE models based on features selection+SOMTE data augmentation.

Model	Balanced accuracy	Accuracy	Recall	FPR	F1-score	RMSE	AUC
RF	0.016	0.016	0.018	0.007	0.016	0.012	0.014
SVM	0.012	0.012	0.018	0.010	0.013	0.025	0.014
NN	0.010	0.011	0.010	0.007	0.011	0.012	0.011
SE	0.007	0.008	0.007	0.004	0.007	0.009	0.006

In summary, it can be observed that the SE and NN models consistently exhibit superior performance compared to the other models across all tables. The findings additionally indicate that the utilization of feature selection and data augmentation techniques can enhance the performance of models and mitigate the variability in their performance. However, the optimal selection of a model is contingent upon the specific problem at hand and the characteristics of the dataset.

4.4. Statistical Test Using Wilcoxson Test.

The Wilcoxon test, also called the Wilcoxon signed-rank test, is a non-parametric statistical test used to compare two similar groups. It is often used when the data doesn't fit the t-test's standards, such as being normal or having the same number of high and low scores. The Wilcoxon test ranks the differences between the two sets of data and checks to see if the positive and negative ranks are spread out in the same way around the median.

The Wilcoxon test can be written in math as:

$$W = \sum_{i=1}^{n} rsign(x_i - y_i) \times rank(|x_i - y_i|) \qquad (10)$$

Where W is the test statistic, x and y are the paired samples, n is the number of pairs, $sign(x_i - y_i)$ is the sign of the difference between the *ith* pair, and $rank(|x_i - y_i|)$ is the rank of the absolute value of the difference between the ith pair.

The null hypothesis of the Wilcoxon test is that the median difference between the paired samples is zero. Suppose the p-value associated with the test statistic is less than the significance level, typically 0.05. In that case, we reject the null hypothesis and conclude that there is a significant difference between the paired samples. The Wilcoxon test is widely used in medicine, psychology, and economics, where paired samples are common. It serves as a robust alternative to the t-test and can be employed for hypothesis testing without making assumptions about the underlying distribution of the data. However, when the sample size is small, it may have less power than the t-test.

Table 10 presents the results of the Wilcoxon signed-rank test comparing the performance of four ML models-RF, SVM, NN and SE-based on balanced accuracy using different feature selection techniques.

The table displays the p-values for pairwise comparisons between the SE model and each of the other three models under each feature selection scenario.

The results suggest that the SE model significantly outperforms the RF model when no feature selection or only SOMTE feature selection is applied. However, the SE model performs significantly worse than the SVM and NN models across all feature selection scenarios. The p-values imply that the differences in performance between the SE model and the other three models are statistically significant in most cases, indicating that the SE model may not always be the optimal choice for classification tasks. Overall, the results emphasize the importance of selecting an appropriate ML algorithm and feature selection technique based on the specific characteristics of the data and the task at hand.

Table 10. P-values for pairwise comparisons of 4 ML models based on balanced accuracy using the Wilcoxon signed-rank test.

Comparison	No features selection, No and SOMTE	No features selection and SOMTE	Features selection, and SOMTE
Ensemble vs. RF	p-value>=0.05	p-value<0.05	p-value<0.05
Ensemble vs. SVM	p-value>=0.05	p-value>=0.05	p-value<0.05
Ensemble vs. NN	p-value>=0.05	p-value>=0.05	p-value<0.05

Table 11 presents the results of a Wilcoxon signed-rank test comparing the performance of three SE models based on balanced accuracy. The models have different configurations regarding feature selection, NO, and SOMTE. The models are denoted as Model A, B, and C for easier understanding.

- Model A: NO features selection, NO, and SOMTE
- Model B: features selection, NO, and SOMTE
- Model C: features selection and SOMTE

The results of the test show that there are statistically significant differences between each pair of models:

- Model A vs. Model B: the p-value is less than 0.05, indicating a significant difference between these two models.
- Model A vs. Model C: the p-value is less than 0.05, suggesting a significant difference between these models.
- Model B vs. Model C: again, the p-value is less than 0.05, indicating a significant difference between these models.
- In summary, the table results demonstrate that all three comparisons show statistically significant differences in the performance of the SE models based on balanced accuracy, as evidenced by the p-values being less than 0.05.

To sum up, the ML-based approach using SE with SMOTE has demonstrated the potential to enhance the accuracy and robustness of malware detection in robotic networks. Its strengths lie in addressing the class imbalance, leveraging the capabilities of multiple ML models, and mitigating overfitting. Nevertheless, this approach may encounter difficulties when dealing with high-dimensional datasets and demand considerable computational resources. Future research could investigate incorporating deep learning models and sophisticated feature selection techniques to boost the performance of the proposed approach.

Table 11. P-values comparisons of three experiments SE models based on balanced accuracy using the Wilcoxon SE models based on balanced accuracy using the Wilcoxon signed-rank test.

Comparison	Model Configuration	Result
Model A vs. Model B	NO features selection, NO, and SOMTE vs. features selection, NO, and SOMTE	<0.05
Model A vs. Model C	NO features selection, NO, and SOMTE vs. features selection and SOMTE	<0.05
Model B vs. Model C	features selection, NO, and SOMTE vs. features selection and SOMTE	<0.05

5. Conclusions

In conclusion, this research introduces a novel approach to malware detection in robotic systems, employing SE with SMOTE data augmentation. This innovative method tackles the challenges of class imbalance, high dimensionality, data heterogeneity, and balancing detection accuracy with false positive rates. The assessment of a publicly accessible robotic systems dataset demonstrated that the proposed approach surpasses individual models and existing techniques regarding detection accuracy and false positive rates. Moreover, the study underscores the significance of incorporating ensemble methods and data augmentation for enhancing the performance and robustness of ML-based approaches in robotic malware detection. Future research could delve into integrating alternative ensemble methods and feature selection techniques to optimize the performance of the proposed approach. We also employed the Wilcoxon signed rank and stability selection tests to verify the proposed approach's performance and ensure its reliability and stability. The results of these tests confirm that the proposed approach is both statistically significant and stable. Further research avenues could involve incorporating other ensemble methods and feature selection techniques, such as deep ensembles, to bolster the proposed approach's performance.

Acknowledgements

The authors would like to thank the Deanship of Scientific Research at the University of Petra for Supporting this Research.

References

[1] Akpinar K. and Ozcelik I., “Analysis of Machine Learning Methods in EtherCAT-Based Anomaly Detection,” *IEEE Access*, vol. 7, pp. 184365-

184374, 2019. DOI:10.1109/ACCESS.2019.2960497

[2] Alamer A. and Basudan S., "Security and privacy of Network Transmitted System in the Internet of Robotic Things," *The Journal of Supercomputing*, vol. 78, no. 16, pp. 18361-18378, 2022. DOI:10.1007/s11227-022-04612-2

[3] Alheeti K., Al-Zaidi R., Woods J., and McDonald-Maier K., "An Intrusion Detection Scheme for Driverless Vehicles Based Gyroscope Sensor Profiling," *in Proceedings of the IEEE International Conference on Consumer Electronics*, Las Vegas, pp. 448-449, 2017. DOI:10.1109/ICCE.2017.7889391

[4] Al-Slais Y. and Ali M., "Robotic Process Automation and Intelligent Automation Security Challenges: A Review," *in Proceedings of the International Conference on Cyber Management and Engineering*, Bangkok, pp. 71-77, 2023. DOI:10.1109/CyMaEn57228.2023.10050996

[5] Breiman L., "Random Forests," *Machine Learning*, vol. 45, pp. 5-32, 2001. https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf

[6] Chawla N., Bowyer K., Hall L., and Kegelmeyer W., "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. https://arxiv.org/pdf/1106.1813.pdf

[7] Clark G., Doran M., and Glisson W., "A Malicious Attack on the Machine Learning Policy of a Robotic System," *in Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering*, New York, pp. 516-521, 2018. DOI:10.1109/TrustCom/BigDataSE.2018.00079

[8] Gao Y., Sun G., Liu J., Shi Y., and Wu L., "State Estimation and Self-Triggered Control of CPSs against Joint Sensor and Actuator Attacks," *Automatica*, vol. 113, pp. 108687, 2020. https://doi.org/10.1016/j.automatica.2019.108687

[9] Han S., Xie M., Chen H., and Ling Y., "Intrusion Detection in Cyber-Physical Systems: Techniques and Challenges," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1052-1062, 2014. DOI:10.1109/JSYST.2013.2257594

[10] Hector J., Katsiaris P., Carey N., Cote N., and Rawat D., "On the Security of Cyber-Physical Robotic Systems Using Dynamic Modeling and Simulation," *in Proceedings of the IEEE International Conference on Communications Workshops*, Montreal, pp. 1-6, 2021. DOI:10.1109/ICCWorkshops50388.2021.9473818

[11] Hong J., Liu C., and Govindarasu M., "Integrated Anomaly Detection for Cyber Security of the Substations," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1643-1653, 2014. DOI:10.1109/TSG.2013.2294473

[12] Jiang J. and Chen Y., "Industrial Control System Anomaly Detection and Classification Based on Network Traffic," *IEEE Access*, vol. 10, pp. 41874-41888, 2022. DOI:10.1109/ACCESS.2022.3167814

[13] Kaur U., Zhou H., Shen X., Min B., and Voyles R., "RoboMal: Malware Detection for Robot Network Systems," *in Proceedings of the 5th IEEE International Conference on Robotic Computing*, Taichung, pp. 65-72, 2021. DOI:10.1109/IRC52146.2021.00016

[14] Khojasteh M., Khina A., Franceschetti M., and Javidi T., "Learning-Based Attacks in Cyber-Physical Systems," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 437-449, 2021. DOI:10.1109/TCNS.2020.3028035

[15] Koren I., "Detecting and Counteracting Benign Faults and Malicious Attacks in Cyber Physical Systems," *in Proceedings of the 7th Mediterranean Conference on Embedded Computing*, Budva, pp. 2-2, 2018. DOI:10.1109/MECO.2018.8405951

[16] Li W., Xie L., and Wang Z., "Two-Loop Covert Attacks Against Constant Value Control of Industrial Control Systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 663-676, 2019. DOI:10.1109/TII.2018.2819677

[17] Marchang J. and Di Nuovo A., "Assistive Multimodal Robotic System (AMRSys): Security and Privacy Issues, Challenges, and Possible Solutions," *Applied Sciences*, vol. 12, no. 4, p. 2174, 2022. https://doi.org/10.3390/app12042174

[18] Martín F., Soriano E., and Canas J., "Quantitative Analysis of Security in Distributed Robotic Frameworks," *Robotics and Autonomous Systems*, vol. 100, pp. 95-107, 2018. https://doi.org/10.1016/j.robot.2017.11.002

[19] Maushart F., Prorok A., Hsieh M., and Kumar V., "Intrusion Detection for Stochastic Task Allocation in Robot Swarms," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, pp. 1830-1837, 2017. DOI:10.1109/IROS.2017.8205998

[20] Mousavinejad E., Ge X., Han Q., Yang F., and Vlacic L., "Resilient Tracking Control of Networked Control Systems Under Cyber Attacks," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 2107-2119, 2021. DOI:10.1109/TCYB.2019.2948427

[21] Narayanan V. and Bobba R., "Learning Based Anomaly Detection for Industrial Arm Applications," *in Proceedings of the Workshop on Cyber-Physical Systems Security and PrivaCy*, Toronto, pp. 13-23, 2018. https://doi.org/10.1145/3264888.3264894

[22] Pawar K., Dharwadkar N., Deshpande P., Honawad S., and Dharmadhikari P., "An Android Based Smart Robotic Vehicle for Border Security Surveillance System," *in Proceedings of the 4th International Conference on Computational Intelligence and Communication Technologies*, Sonepat, pp. 296-301, 2021. DOI: 10.1109/CCICT53244.2021.00062

[23] Pang Z., Liu G., Zhou D., Hou F., and Sun D., "Two-Channel False Data Injection Attacks against Output Tracking Control of Networked Systems," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3242-3251, 2016. DOI: 10.1109/TIE.2016.2535119

[24] Pu H., He L., Cheng P., Sun M., and Chen J., "Security of Industrial Robots: Vulnerabilities, Attacks, and Mitigations," *IEEE Network*, vol. 37, no. 1, pp. 111-117, 2023. DOI:10.1109/MNET.116.2200034

[25] Sharifi S., Usman M., and Gul E., "An Intelligent Health Control Security Robotic System," *University of Wah Journal of Computer Science*, vol. 4, no. 1, pp. 17-30, 2022. https://uwjcs.org.pk/index.php/ojs/article/view/55

[26] Singh R., Kushwah A., Warrier P., and Oza S., "Wireless Surveillance Robot for Industrial Application," *in Proceedings of the Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND*, Singapore, pp. 561-573, 2021. https://doi.org/10.1007/978-981-19-5868-7_41

[27] Souza L., Rocha F., and Soares M., "A Review on Software/Systems Architecture Description for Autonomous Systems," *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, vol. 16, no. 3, pp. 52-60, 2023. DOI:10.2174/2666255815666220513101350

[28] Sun Y., Shao H., and Zhang B, "Ensemble Based on Accuracy and Diversity Weighting for Evolving Data Streams," *The International Arab Journal of Information Technology*, vol. 19, no. 1, pp. 90-96, 2022. https://doi.org/10.34028/iajit/19/1/11

[29] Tang Y., Zhang D., Ho D., Yang W., and Wang B., "Event-Based Tracking Control of Mobile Robot with Denial-of-Service Attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 9, pp. 3300-3310, 2020. DOI:10.1109/TSMC.2018.2875793

[30] Zhao Z., Huang Y., Zhen Z., and Li Y., "Data-Driven False Data-Injection Attack Design and Detection in Cyber-Physical Systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 6179-6187, 2021. DOI:10.1109/TCYB.2020.2969320

[31] Zhou C., Hu B., Shi Y., Tian Y., Li X., and Zhao Y., "A Unified Architectural Approach for Cyberattack-Resilient Industrial Control Systems," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 517-541, 2021. DOI:10.1109/JPROC.2020.3034595

[32] Zhou C., Zhou C., Huang S., Xiong N., Yang S., Li H., Qin Y., and Li X., "Design and Analysis of Multimodel-Based Anomaly Intrusion Detection Systems in Industrial Process Automation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1345-1360, 2015. DOI:10.1109/TSMC.2015.2415763

Ali Al Maqousi is currently the director of Office of External Funding and Technology Transfer at University of Petra and an assistance professor at the Faculty of Information Technology, Amman-Jordan. He was appointed head of the department of Computer Science and Computer Networks during the period of Sep. 2009 till Sep. 2012. He received his PhD in computer science from Oxford Brookes University, UK, 2003 for his work on providing Quality of Service (QoS) in packet switched networks. He published more than twenty articles in international journals and conferences. He organized and co-organized four IEEE international conferences in ICT and its applications. He managed and coordinated several joint funded projects in different topics such as Reconow, ENEPLAN, TEJ, BittCoin, FREE, Neucare Projects. He is involved in research relating to multi-service networking, network performance, security and privacy, social networks, smart cities and smart grids. He is a member of several Erasmus+CBHE and ICM projects. He is an IEEE senior member.

Mohammad Alauthman received his Ph.D. from Northumbria University Newcastle, the UK 2016. He received a B.Sc. degree in Computer Science from Hashemite University, Jordan, in 2002 and an M.Sc. in Computer Science from Amman Arab University, Jordan, in 2004. He is an Assistant Professor at the Information Security Department at Petra University, Jordan. His research interests include Cyber-Security, Cyber Forensics, Advanced Machine Learning and Data Science applications.