Trust Score Prediction for IoT Device Onboarding Using Transfer and Few-Shot Learning in Consumer Electronics

Ilias Politis[1], Michail Bampatsikos[2], Apostolis Zarras[2] and Christos Xenakis[2]

[1]Industrial Systems Institute, ATHENA Research Center, Patras, Greece
[2]Systems Security Laboratory, University of Piraeus, Greece

***Abstract*—The rapid proliferation of Internet of Things (IoT) devices in consumer electronics has made efficient trust score prediction essential for secure device onboarding. This paper presents a hybrid Trust Management framework that integrates few-shot and transfer learning with a statistical Markov chain foundation to address data scarcity and adaptability challenges in dynamic IoT environments. The few-shot learning phase enables rapid adaptation from minimal data (as few as 5–20 onboarding samples), while transfer learning ensures robust cross-domain generalizability (e.g., from consumer to industrial IoT). Comprehensive evaluation demonstrates that, with 20 onboarding samples, the proposed approach achieves fine-tuned *Mean Squared Error* (MSE) as low as 4.45 (XGBoost) and 5.07 (Random Forest), R^2 scores exceeding 0.96, and average prediction error (MAE) below 1.55. Batched distributed ledger operations reduce total onboarding latency to under 750 ms for five devices, with system throughput averaging 137.7 devices per second across models. These results show the framework delivers accurate, low-latency, and scalable trust score predictions suitable for real-time onboarding in evolving IoT environments.**

***Index Terms*—IoT, Trust Score Prediction, Transfer Learning, Few-Shot Learning, Device Onboarding, Real-Time Prediction**

I. Introduction

The rapid proliferation of *Internet of Things* (IoT) devices in consumer electronics has introduced significant challenges in ensuring secure and reliable device interactions, particularly during onboarding in dynamic, data-constrained environments. A robust *Trust Management* (TM) layer is indispensable for rapidly gauging a newcomer's reliability, shielding the network from data-poisoning and zero-day threats, and achieving these objectives within the strict data and resource constraints typical of consumer-grade IoT. However, existing TM approaches present notable limitations. Statistical models, such as our prior Markov chain-based *Multi Attribute Decision Making* (MADM) framework [1], are robust and resource-efficient but exhibit limited adaptability to rapidly evolving threats and new attack vectors. Conversely, purely *Machine Learning* (ML)-based methods [2] often require large training datasets, limiting their practicality in real-time onboarding scenarios—such as in smart homes, healthcare wearables, and industrial IoT—where only minimal data is available for new devices.

This motivates the need for a *hybrid approach* that preserves the efficiency and interpretability of statistical models while incorporating the adaptability of ML techniques under extreme data scarcity. The central problem addressed in this paper is therefore: *how to reliably predict the trustworthiness of a new IoT device during its onboarding process when only a handful of labelled samples (5–20) are available, while meeting strict latency budgets (sub-500 ms) and ensuring resilience against adversarial manipulation*. Solving this problem requires not only accurate trust score predictions under limited data but also system-level scalability to handle large device populations in real time.

To address these challenges, this paper proposes a novel hybrid TM framework that integrates: (i) *Few Shot Learning* (FSL) for rapid adaptation from minimal onboarding data; (ii) transfer learning for cross-domain generalisation (e.g., consumer-to-industrial IoT); and (iii) a foundational Markov chain-based statistical model for interpretable and deterministic trust estimation. The novelty of our approach lies in this explicit integration: unlike prior TM solutions, we combine the statistical robustness of a Markov/MADM trust function with the adaptability of ML predictors and enhance it further with a mathematically justified blending strategy between base and fine-tuned models. This makes the framework resilient to poisoning of scarce onboarding samples while ensuring adaptability to device-specific behaviours.

Our experimental evaluation, using Random Forest, k-NN, and XGBoost as trust predictors, demonstrates that the proposed framework achieves fine-tuned *Mean Squared Error* (MSE) as low as 4.45 (XGBoost) and 5.07 (Random Forest), R^2 scores exceeding 0.96, and *Mean Absolute Error* (MAE) below 1.55 with 20 onboarding samples. The approach achieves system throughput up to 175 devices per second (k-NN) and consistently averages above 137 devices/s across models. Onboarding latency is reduced to under 750 ms for five devices using batched distributed ledger (DLT) operations. To further improve prediction accuracy and adaptability under extreme data scarcity, we employ a mathematically justified blending of the base and fine-tuned models, with optimal weights (α) empirically determined via grid search to minimise validation error. This blended prediction strategy ensures both robustness and rapid domain adaptation, as validated by detailed ablation and sensitivity analyses.

To our knowledge, this is the first TM framework to explicitly integrate few-shot learning, transfer learning, and a Markov chain-based trust estimation model for secure IoT device onboarding. The contributions of this paper can be summarised as follows:

- Problem formulation: we formally define the challenge of trust score prediction under scarce onboarding data and

strict latency constraints, a gap not directly addressed in existing TM research.
- Novel hybrid framework: we introduce the first TM framework that fuses statistical Markov-based modelling, transfer learning, and few-shot adaptation, augmented with a principled blending mechanism.
- Scalable, low-latency evaluation: we demonstrate experimentally that the framework achieves high accuracy, sub-second onboarding, and throughput exceeding 137 devices/s, validating its applicability to real-world consumer electronics.

The remainder of the paper is organised as follows. Section II reviews related work on TM in IoT. Section III introduces the algorithmic design and logic. Section IV presents performance evaluation using real-world use cases. Section V discusses comparative advantages and security benefits. Section VI concludes the paper and outlines future research directions.

II. Literature Review

This literature review surveys key TM approaches in IoT networks, focusing on methodologies used to compute device trust scores based on factors such as *Quality of Service* (QoS), delays, security, and feedback. By examining existing solutions, we underscore how the proposed hybrid framework advances the state of the art. Special emphasis is placed on our prior work leveraging a Markov chain-based method [1], a critical foundational element in this research.

A. Machine Learning Approaches in Trust Management

ML techniques have been extensively employed in IoT trust management. For instance, Shayesteh et al. [3] apply Bayesian learning and Dempster-Shafer Theory to derive trust scores from entity and data reliability. Alghofaili and Rassam [4] present a Multi-Criteria Decision-Making approach integrated with a *Deep Long Short-Term Memory* (LSTM) model to evaluate trust through packet loss and throughput metrics, weighted by Shannon's entropy. Researchers have also proposed a Federated Learning-based TM framework for Industrial IoT, where trust is calculated via a weighted sum [5]. Furthermore, Wang et al. [6] employ unsupervised learning and clustering to differentiate trustworthy vehicles in IoV, using Direct Trust and Indirect Trust. Meanwhile, Ma et al. [7] leverage LSTM to predict device behaviour by incorporating QoS, a measure of network performance, delays, security, and feedback, and time-dependent features.

Despite providing valuable insights, these ML-based methods often depend on large datasets, limiting their effectiveness in data-scarce IoT settings. In addition, they tend to focus on a narrow range of metrics, reducing their broader applicability.

B. Non-Machine Learning Approaches in Trust Management

Non-ML TM methods rely on statistical or structural techniques. Alam et al. [2] propose a TM method that calculates trust scores using QoS and cooperation metrics. Latif [8] introduces a context-dependent approach for social IoT, integrating device capabilities and satisfaction. Bampatsikos et al. [1] present a two-dimensional Markov chain model, combined with MADM and a piecewise function, to predict trust evolution based on cyber risk, packet loss, and device utilisation. Bampatsikos et al. [9] also explore this approach with Hyperledger Fabric. This offers a strong statistical foundation but struggles with dynamic threats due to its static nature. Other works, such as Liu et al. [10] (i.e., Hidden Markov Model for VANETs) and Bai et al. [11] (i.e., game theory in supply chains), provide stability yet lack adaptability to new or evolving scenarios.

C. Beyond the State of the Art

The reviewed literature highlights limitations in both ML- and non-ML-based TM approaches. ML methods require large training datasets, potentially diminishing efficiency [7], whereas non-ML approaches tend to be less adaptable to emerging threats [1]. Our prior Markov-based TM method provides a robust baseline by employing statistical modelling of key parameters such as cyber risk and QoS, but its predefined rules impede responsiveness in dynamic IoT environments [1].

The proposed hybrid framework addresses these issues by integrating transfer learning and FSL. Transfer learning leverages pre-trained models on a synthetic Markov chain dataset, ensuring efficiency and enabling cross-domain adaptability (e.g., from consumer to industrial IoT). Meanwhile, FSL enhances this framework by requiring minimal data (only five samples), facilitating rapid trust assessment (e.g., XGBoost inference at 0.9425 ms) with an average prediction error of 2.29. Unlike resource-intensive ML methods, the solution avoids large dataset dependencies, and unlike purely statistical models, it remains adaptable to evolving threats. Moreover, the statistical ground truth mitigates poisoning attacks by building upon the Markov foundation while resolving its inherent limitations. Consequently, the proposed framework significantly advances consumer IoT security and scalability.

III. Algorithm Design and Logic

The proposed framework for trust score prediction in IoT device onboarding leverages transfer learning and few-shot learning techniques to address the pervasive challenge of sparse data in consumer electronics environments. Building on a previous work which employed a 2D Markov chain model to simulate trust dynamics [1], this study introduces a novel hybrid ML and statistical approach for predicting trust scores in new IoT devices. By enabling secure, real-time onboarding in resource-constrained consumer IoT contexts, such as wearable health devices and autonomous vehicle systems, this method ensures the rapid trust assessment necessary for operational safety and security. This section elucidates the conceptual design logic of the proposed algorithm and provides a mathematically grounded description of its implementation, including feature engineering, model training, and prediction blending.

A. Design Logic

The proposed trust score prediction algorithm is driven by three primary challenges in consumer IoT device onboarding:

(i) the scarcity of labeled data for new devices, (ii) the need for rapid trust assessment to facilitate real-time decision-making, and (iii) the requirement for resilience against security threats (e.g., data poisoning, zero-day attacks). To address these issues, the algorithm employs a two-phase structure that combines transfer learning (to exploit prior knowledge) with FSL (to adapt efficiently to limited onboarding data).

1) Few-Shot Transfer Learning: Concept and Rationale: Few-shot transfer learning synthesises the benefits of transfer learning and FSL to achieve effective and flexible trust score prediction. Transfer learning relies on a pre-trained model, initially trained on a large, heterogeneous dataset. We utilise synthetic data from a 2D Markov chain model [1], providing a robust foundation for trust prediction. Few-shot learning then fine-tunes this model using a small number of onboarding samples (e.g., 5 samples) to adapt to the specific characteristics of new IoT devices. This hybrid approach ensures that the model can generalise from substantial prior knowledge while adapting rapidly to sparse, real-time data, a critical requirement for consumer electronics applications.

Several factors motivated this approach. First, consumer IoT onboarding faces severe data limitations from resource constraints, privacy regulations, or the novelty of emerging devices (e.g., next-generation health wearables or autonomous vehicle modules). Few-shot learning addresses this issue by requiring only a small number of samples, thereby mitigating data collection overhead. Second, real-time operation in consumer IoT environments demands swift trust evaluation. Transfer learning accelerates this process by providing a robust starting point by reducing training time, while few-shot fine-tuning ensures rapid adaptation without requiring comprehensive retraining. Third, the approach reinforces security and robustness by leveraging a large, diverse training corpus. This pre-training promotes resilience against noise and attacks, while few-shot fine-tuning tailors the model to specific device attributes, mitigating overfitting risks on small datasets.

This methodology yields multiple advantages. It offers *efficiency* by reducing data and computational demands; a pivotal factor for resource-constrained IoT devices. It provides *adaptability* by enabling the model to accommodate new devices using minimal samples; an important capability for dynamic consumer IoT ecosystems. It also enhances *accuracy and stability* by employing a blended prediction strategy, achieving high performance (e.g., a fine-tuned *Random Forest* (RF) yields $R^2 = 0.8891$). Finally, it enhances *security* via reliance on statistical ground truth and blending predictions, reducing data poisoning in few-shot samples, a common security concern in IoT contexts.

B. Algorithm Description

The implementation of the algorithm formalises this design logic in mathematical terms. The framework proceeds in three stages: transfer learning using a Markov-based statistical ground truth, few-shot fine-tuning with limited onboarding data, and blended prediction for robust decision-making. Figure 1 illustrates the workflow, while Algorithm 1 provides pseudocode. The mathematical representation of each stage is presented below.

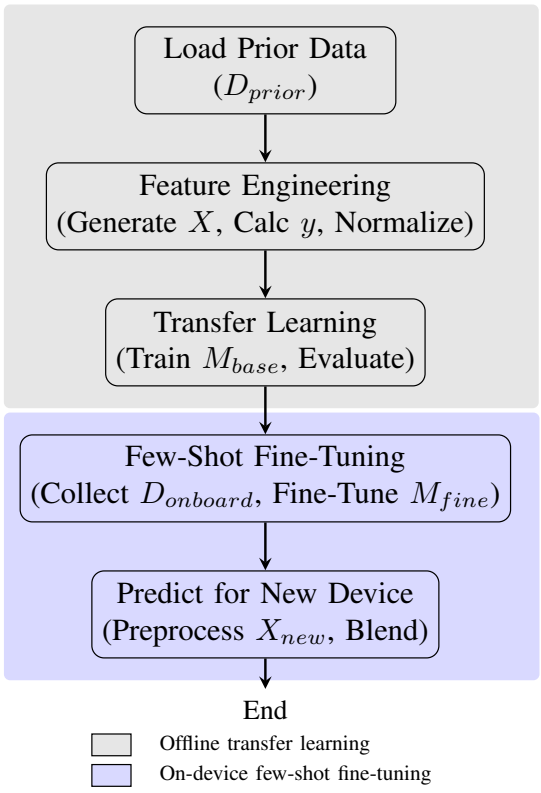

Fig. 1. End-to-end workflow of the proposed hybrid trust-score engine. The upper band (grey) represents offline transfer learning; the lower band (blue) depicts on-device few-shot fine-tuning. Their convergence shows how each inference blends long-term knowledge with real-time adaptation. The final stage includes latency evaluation (Eq. 10), where onboarding delay is decomposed into identity, communication, ledger, and inference components to ensure compliance with the 500 ms real-time target.

*1) **Statistical Ground Truth and Feature Representation**:* Each IoT device i is described by a feature vector

$$x_i = \{C_i, R_i, S_i, P_i, RS_i, RepS_i, OS_i, Pkg_i\}, \qquad (1)$$

where C_i, R_i denote CPU and RAM utilisation, S_i is the security level, P_i the packet loss ratio, $RS_i, RepS_i$ the risk and reputation states, and (OS_i, Pkg_i) the inverted ages of the operating system and installed packages. The problem is to learn a regression function

$$f : \mathbb{R}^8 \to [0, 100], \quad f(x_i) \approx T_i,$$

where T_i is the trust score that quantifies the device's reliability during onboarding.

a) Ground Truth Trust Score: The statistical trust score is generated via the piecewise function:

$$T = \begin{cases} \max(0, \min(100, 100 - 20R - 10C + \\ +4S + 2OS_i + 1P_i - 10P)), & \text{if } RS \geq 4 \\ \min(100, 80 - 20R - 10C + \\ +4S + 2OS_i + 1P_i - 10P), & \text{if } RS \leq 2 \\ \max(20, \min(100, 70 - 20R - 10C + \\ +4S + 2OS_i + 1P_i - 10P)), & \text{otherwise} \end{cases} \qquad (2)$$

Equation 2 balances negative contributions from utilisation and packet loss with positive contributions from security and freshness. The coefficients (e.g., 20, 10, 4) are empirically derived from domain knowledge and prior analysis [1].

Algorithm 1 IoT Device Onboarding Trust Score Prediction

Input: Pre-generated Markov chain data (D_{prior}), onboarding samples ($D_{onboard}$), device features (X_{new})
Output: Predicted trust score (T_{final}) for a new device
Load Prior Data: Load $D_{prior} = \{RepState, RiskState, TrustScores\}$ from CSV files.
Feature Engineering:
  a. Generate synthetic features $X = \{C, R, S, P, RS, RepS, OS_i, P_i\}$.
  b. Calculate trust scores y using Eq. 2.
  c. Normalize X using MinMaxScaler to obtain X_{scaled}.
Transfer Learning (Base Training):
  a. Split X_{scaled}, y into training (80%) and test (20%) sets.
  b. Train base models $M_{base} = \{RF, k\text{-}NN, XGBoost\}$ by minimising Eq. 3.
  c. Evaluate M_{base} on test set, computing metrics (MSE, R^2, MAE, MAPE, RMSE, inference time).
Few-Shot Fine-Tuning:
  a. Collect $D_{onboard} = \{X_{onboard}, y_{onboard}\}$ with K samples.
  b. Impute missing features in $X_{onboard}$ with training set means, normalize to $X_{onboard,scaled}$.
  c. Fine-tune models $M_{fine} = \{RF_{fine}, k\text{-}NN_{fine}, XGBoost_{fine}\}$ by minimising Eq. 5.
  d. Blend predictions with Eq. 6:

$$T_{final} = \alpha \cdot M_{base}(X_{test}) + (1 - \alpha) \cdot M_{fine}(X_{test}).$$

  e. Evaluate blended predictions on test set, computing metrics.
Prediction for New Device:
  a. Preprocess X_{new}: impute missing features, normalize to $X_{new,scaled}$.
  b. Predict $T_{base} = M_{base}(X_{new,scaled})$, $T_{fine} = M_{fine}(X_{new,scaled})$.
  c. Compute $T_{final} = \alpha \cdot T_{base} + (1 - \alpha) \cdot T_{fine}$ per Eq. 6.
  d. Return T_{final}.
System Latency Evaluation:
  Compute end-to-end onboarding latency L according to Eq. 10, decomposing identity (L_{SSI}), messaging (L_{MQTT}), ledger (L_{DLT}), and inference (L_{inf}) components.

Importantly, Eq. 2 also encodes domain knowledge by penalising resource saturation (C_i, R_i) and packet loss (P_i) with relatively high weights, since these directly degrade device reliability. Positive contributions are drawn from security level (S_i) and software freshness (OS_i, Pkg_i), which improve resilience and reduce vulnerability. The bounding functions (*min*, *max*) ensure that scores remain within $[0, 100]$, preventing extreme fluctuations due to any single feature. This design reflects practical IoT considerations (i.e., no device is entirely untrustworthy if its core behaviour is stable, and no device is fully trustworthy if critical resources are exhausted).

b) Markov Dynamics of States: As it was detailed in [1], the risk (RS) and reputation ($RepS$) states evolve over time according to a two-dimensional Markov chain model [1]. Let $p^{(t)}$ denote the joint distribution of $(RS, RepS)$ at time t. The state transitions follow:

$$p^{(t+1)} = p^{(t)}\mathbf{P},$$

where $\mathbf{P}$ is the joint transition matrix. This captures temporal correlations in device behaviour, e.g., a device with declining reputation is more likely to remain in low-trust states, while a device with strong historical reputation tends to persist in higher-trust states.

c) Feature Sampling: The remaining features are stochastically sampled to reflect heterogeneous device characteristics. CPU and RAM utilisation are drawn from truncated uniform distributions,

$$C_i \sim \mathcal{U}(0.2, 0.8), \quad R_i \sim \mathcal{U}(0.1, 0.9),$$

security level is sampled from a categorical distribution biased towards medium security,

$$S_i \sim \text{Categorical}(\{1, 2, 3, 4, 5\}, p_S), \quad p_S = (0.2, 0.5, 0.3),$$

packet loss follows a Beta distribution skewed towards low-loss regimes,

$$P_i \sim \text{Beta}(\alpha = 2, \beta = 8),$$

and OS/package ages are sampled uniformly as inverted values to reflect freshness:

$$OS_i, Pkg_i \sim \mathcal{U}\{1, 2, 3, 4, 5\}.$$

d) Synthetic Dataset.: The labelled dataset is then

$$\mathcal{D} = \{(x_i, T_i)\}_{i=1}^{N}, \quad N = 1000,$$

with T_i computed from Eq. 2. This setup encodes domain knowledge such as resource saturation (C_i, R_i) and packet loss (P_i) receive heavy penalties, while higher security levels (S_i) and updated software (OS_i, Pkg_i) improve trust. The resulting dataset provides both trusted and untrusted devices in realistic proportions (approximately 64% trusted, 36% untrusted), ensuring balanced training for transfer learning and robust evaluation.

*2) **Transfer Learning Phase**:* The base models are trained on $\mathcal{D}$ by minimising mean squared error:

$$f_b = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - T_i)^2, \tag{3}$$

where $f_b \in \{f_{\text{RF}}, f_{\text{kNN}}, f_{\text{XGB}}\}$. RF provides robustness, k-NN simplicity, and XGBoost high inference efficiency. These models encode transferable priors.

The optimisation in Eq. 3 produces base models f_b that serve as priors: they learn a generic mapping from the feature space to trust scores across a wide range of devices. Training on synthetic data allows these models to capture general trust dynamics without relying on scarce real-world onboarding samples. Random Forest provides robustness against noise and non-linear feature interactions, k-NN offers sensitivity to local variations in feature space and XGBoost contributes high accuracy with millisecond-level inference time. Importantly, each model is trained and evaluated independently, not as an ensemble, to allow systematic comparison of strengths and weaknesses under sparse onboarding conditions.

*3) **Few-Shot Fine-Tuning Phase**:* For onboarding, only K samples are available per device:

$$\mathcal{D}_{few} = \{(x_j, T_j)\}_{j=1}^{K}, \quad K \in \{5, 10, 20\}. \tag{4}$$

The fine-tuned model is trained with reduced complexity to avoid overfitting:

$$f_f = \arg\min_{f \in \mathcal{F}} \frac{1}{K} \sum_{j=1}^{K} (f(x_j) - T_j)^2. \tag{5}$$

The dataset $\mathcal{D}_{few}$ contains only K samples per device, with $K \in \{5, 10, 20\}$ chosen to reflect realistic onboarding conditions where only a handful of interactions are available before the device must be trusted or rejected. The optimisation in Eq. 5 adapts the base model to these limited samples. To reduce variance under such small K, the fine-tuned models are deliberately restricted in complexity (e.g., fewer trees in RF, shallower depth in XGBoost). Unlike typical N-way K-shot classification settings, our formulation is regression hence, each sample provides a continuous trust score label and K indicates the number of available regression pairs. This ensures the model adapts without overfitting to a handful of points.

*4) **Blended Prediction**:* To mitigate the risk of overfitting in few-shot scenarios while ensuring adaptability, the framework employs a blended prediction strategy that combines the base and fine-tuned models. The final trust score for a device is given by:

$$T_{final}(x) = \alpha f_b(x) + (1 - \alpha) f_f(x), \quad \alpha \in [0, 1], \quad (6)$$

where f_b is the base predictor (trained on the large synthetic dataset) and f_f is the fine-tuned predictor (adapted to K onboarding samples).

Equation 6 formalises the trade-off between robustness and adaptability: the base model f_b typically yields low-variance but sometimes biased predictions for new devices, while the fine-tuned model f_f adapts to device-specific behaviour but exhibits higher variance under small K. Blending balances these effects, implementing a bias–variance compromise.

The mathematically optimal solution weight $\alpha^{\backslash *}$ can be obtained by minimising the validation mean squared error (MSE). Let $e_b = f_b(x) - T$ and $e_f = f_f(x) - T$ denote the residuals of the base and fine-tuned models with respect to the ground truth T. The expected blended error is:

$$\mathcal{L}(\alpha) = \mathbb{E}\left[(\alpha e_b + (1 - \alpha) e_f)^2\right]. \quad (7)$$

Expanding and differentiating with respect to α gives:

$$\alpha^{\backslash *} = \frac{S_f - S_{bf}}{S_b + S_f - 2S_{bf}}, \quad (8)$$

where $S_b = \mathbb{E}[e_b^2]$, $S_f = \mathbb{E}[e_f^2]$, and $S_{bf} = \mathbb{E}[e_b e_f]$. In the special case of uncorrelated errors ($S_{bf} \approx 0$), this reduces to:

$$\alpha^{\backslash *} \approx \frac{S_f}{S_b + S_f}, \quad (9)$$

which intuitively assigns more weight to the model with smaller error.

In practice, we compute $\alpha^{\backslash *}$ on a validation split of the few-shot onboarding data. For numerical stability, $\alpha^{\backslash *}$ is clipped to $[0, 1]$ and regularised when the denominator of Eq. 8 is small. When K is very small (e.g., 5), we also evaluate α via grid search to confirm consistency with the analytical solution.

Empirically, $\alpha^{\backslash *} \approx 0.8$ across most experiments, reflecting the dominance of the base model under sparse conditions while still leveraging the corrective signal from the fine-tuned model. This convex combination ensures stable trust estimation without sacrificing adaptability, yielding robust predictions for secure IoT device onboarding.

*5) **Latency Model**:* End-to-end onboarding latency is decomposed as:

$$L = L_{SSI} + L_{MQTT} + L_{DLT} + L_{inf}$$

where $L_{SSI} \approx 50$ ms, $L_{MQTT} \approx 20$ ms, $L_{DLT} \approx 200$ ms, and $L_{inf} < 7$ ms. For K sequential samples:

$$L_{total}(K) \approx K \cdot (L_{SSI} + L_{MQTT} + L_{DLT}) + L_{inf} \quad (10)$$

Batching optimisations reduce L_{DLT} per sample to meet the 500 ms target.

Eq. 10 decomposes end-to-end onboarding latency into four measurable components: identity generation (L_{SSI}), communication (L_{MQTT}), distributed ledger logging (L_{DLT}), and inference time (L_{inf}). Among these, inference latency is consistently below 7 ms and therefore negligible compared to the $\sim$200 ms per-transaction DLT cost. The decomposition makes the bottleneck explicit and motivates batching: aggregating B onboarding events per ledger transaction reduces the amortised L_{DLT} to approximately $200/B$ ms per device, a critical optimisation to meet the 500 ms real-time requirement. This system-level model thus validates the feasibility of the framework under deployment conditions.

Algorithm 1 presents the pseudocode implementation of these stages, directly corresponding to the mathematical framework defined by Equations 2–10.

IV. Use Case and Performance Evaluation

This section presents a practical use case for the proposed trust score prediction algorithm. It details the onboarding scenario for IoT devices in a consumer electronics context. An experimental setup evaluates the implementation of the algorithm. The performance assessment, supported by quantitative metrics and visual aids, illustrates the algorithm's effectiveness and sheds light on its potential for real-world applications, including smart health devices and autonomous vehicles.

A. Use Case and Experimental Setup

The focus of this use case is on the secure onboarding and registration of a health-related smart device (e.g., a fitness tracker) within an IoT network, where a rapid and reliable verification of device credentials is essential for safety and operational trust. The onboarding sequence, as illustrated in Figure 2, starts with the IoT device generating a *Decentralized Identifier* (DID) managed by an *Self-Sovereign Identity* (SSI) management system, adhering to the W3C DID standard [12]. The SSI client then provisions cryptographic keys, offering three roots of trust: *Physical Unclonable Function* (PUF), *Trusted Execution Environment* (TEE), and Hyperledger Aries [13] as a fallback, providing flexibility across diverse hardware platforms.

Upon successful DID creation and publication of the corresponding DID Document, the SSI Manager acknowledges registration. The IoT device then requests issuance of a Verifiable Credential (VC) embedding its unique cryptographic footprint. This credential is signed and the device is paired with its owner, who authenticates the DID using their own keys. Subsequently, the IoT device transmits trust data and its signed

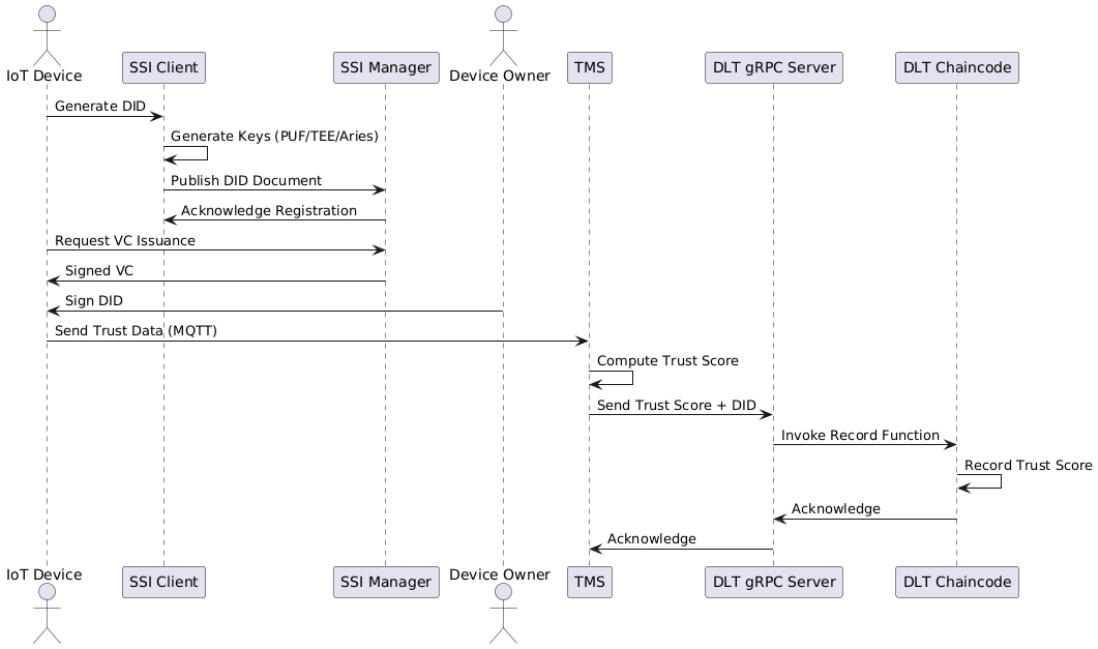

Fig. 2. Secure onboarding message flow. The diagram highlights (a) cryptographic binding of device identity via DID/SSI and (b) tamper-evident logging of the computed trust score to the DLT.

DID to the *Trust Management Server* (TMS) using MQTT. The TMS computes an initial trust score, then records this score and DID to the *Distributed Ledger Technology* (DLT) network via a gRPC interface. The DLT chaincode securely logs the onboarding event, with acknowledgements propagated back to the TMS, ensuring the onboarding completes within the target latency budget.

The full onboarding message flow, including cryptographic binding via SSI and tamper-evident logging to the DLT, is shown in Figure 2.

To evaluate the proposed trust management framework, we simulate a testbed of 50 heterogeneous IoT devices (including wearables and vehicle sensors) using an Intel i7 CPU, 16 GB RAM, and Ubuntu 20.04. Synthetic datasets are generated using the 2D Markov chain-based statistical trust modelling approach from [1], where device trust evolution is a function of both reputation and risk states. Specifically, the dataset consists of 1000 synthetic device samples, each comprising features such as CPU utilisation (C), RAM utilisation (R), security level (S), packet loss (P), risk state (RS), reputation state ($RepS$), and inverted OS/package ages (OS_i, P_i). Trust scores (T) are computed for each device according to the piecewise statistical function defined in Eq. 2.

The use of a Markov chain and MADM-driven synthetic simulation [1] enables the generation of feature vectors and trust scores spanning a wide spectrum of device behaviours and trustworthiness. By explicitly sampling from a range of risk, reputation, and performance states, the simulation provides a dataset that is both sufficiently large and balanced for model training. Consequently, no additional oversampling, undersampling, or post-processing techniques are needed to mitigate data imbalance. When binarising the trust score (e.g., trusted if $T \geq 70$), the dataset includes approximately 64% trusted and 36% untrusted devices, ensuring fair representation for both classes.

In this formulation, resource usage (R, C, P) negatively impacts trust, while security and device/software freshness (S, OS_i, P_i) contribute positively. The risk state thresholds ensure that devices with high risk are penalised, while low-risk states yield higher base trust.

For each experiment, 80% of the synthetic dataset is used for training the base models (Random Forest, k-NN, XGBoost), and 20% for evaluation. Realistic onboarding scenarios are emulated using $K = \{5, 10, 20\}$ few-shot samples per device. All feature values are sampled using Python 3.10, NumPy, and pandas, with truncated normal or beta distributions for CPU, RAM, and packet loss, and categorical assignments for security, risk, and reputation states based on empirically observed device behaviors [1]. The ground truth trust score for each device is generated using the statistical function derived from the Markov chain and MADM-based model. These values serve as the reference labels for supervised training and evaluation of all machine learning models considered in this study. Model predictions are assessed against these Markov-based trust scores to determine accuracy and error metrics throughout our experiments.

The overall pipeline, including model training, fine-tuning, and performance evaluation, is implemented in Python 3.9 with scikit-learn 1.0.2, XGBoost 1.5.0, and Hyperledger Fabric 2.2. The trust scores generated serve as ground truth labels for all supervised learning experiments.

In line with few-shot learning evaluation practices, we adopt a K-shot regression approach in our experimental design. Here, K denotes the number of labelled onboarding samples per device used for model fine-tuning and evaluation, with $K = \{5, 10, 20\}$. Unlike traditional classification-based few-

TABLE I
PERFORMANCE METRICS FOR BASE AND FINE-TUNED MODELS
(FINE-TUNED: 5 ONBOARDING SAMPLES)

Model	MSE	R^2	MAE	Inference Time (ms)
RF (Base)	0.9904	0.9917	0.6796	2.9747
k-NN (Base)	18.3057	0.8460	2.6887	0.3256
XGBoost (Base)	0.4436	0.9963	0.4782	0.4660
RF (Fine-Tuned)	13.1759	0.8891	3.6297	3.4938
k-NN (Fine-Tuned)	33.0026	0.7223	5.7440	3.4716
XGBoost (Fine-Tuned)	20.5551	0.8270	4.5347	0.9425

shot learning (N-way K-shot), our task is formulated as a regression problem for continuous trust score prediction, so the concept of "N-way" (number of classes) does not apply. Instead, we analyse the model's ability to rapidly adapt and generalise from a small set of onboarding data points per device, closely mirroring real-world IoT onboarding scenarios where only limited feature-label pairs are available at runtime.

B. Performance Analysis

The performance of the trust score prediction algorithm is evaluated using quantitative metrics and visual aids, focusing on accuracy, stability, and inference time across the simulated IoT network. The analysis compares RF, *k-Nearest Neighbors* (k-NN), and XGBoost models in both base and fine-tuned configurations, leveraging the few-shot transfer learning approach.

The primary metrics include MSE, R-squared (R^2), MAE, *Mean Absolute Percentage Error* (MAPE), *Root Mean Squared Error* (RMSE), and inference time. These metrics are computed on the test set for base models and the blended predictions for fine-tuned models. Table I summarises the results for the base models and for the fine-tuned models using five onboarding samples, averaged over 10 runs to account for variability.

Table I shows that XGBoost achieves the highest R^2 (0.9963) and lowest MSE (0.4436) among base models, indicating excellent predictive accuracy on the pre-trained dataset. After fine-tuning with only five onboarding samples, all models experience an increase in MSE and a decrease in R^2, consistent with the expected risk of overfitting in extreme few-shot regimes. For example, RF maintains a strong R^2 of 0.8891 but MSE increases to 13.18, while k-NN exhibits the greatest sensitivity to limited data, with R^2 dropping to 0.72 and MSE rising to 33.00. These results reflect the inherent trade-off in few-shot learning: minimal data can limit precision but enable rapid adaptation and personalisation.

To further analyse the impact of onboarding sample size, we systematically varied the number of samples used for fine-tuning. Table II reports the performance of the fine-tuned models as the onboarding sample size increases from 5 to 10 and 20. As shown, increasing the number of onboarding samples substantially improves both MSE and R^2 for RF and XGBoost: for RF, MSE drops from 13.18 (5 samples) to 5.07 (20 samples), while R^2 increases from 0.89 to 0.96. Similarly, XGBoost reaches an MSE of 4.45 and R^2 of 0.96 with 20 samples. k-NN remains less robust in the few-shot regime, though it also benefits from additional data.

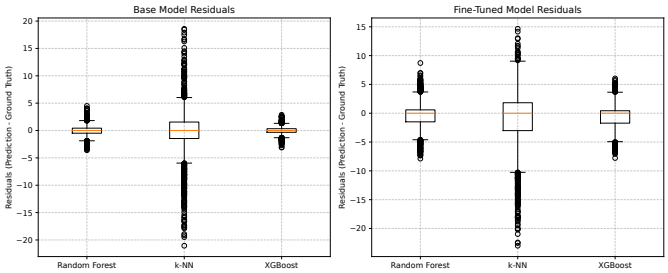

Fig. 3. Residual error distribution for base vs. fine-tuned models (5 onboarding samples). Random Forest maintains stable error distribution with a tight IQR, while k-NN exhibits wider variability and more outliers, reflecting higher sensitivity to sparse data. XGBoost shows moderate dispersion, balancing accuracy and robustness.

These findings demonstrate that, while overfitting can occur when very few samples are available for fine-tuning, the hybrid few-shot transfer learning approach becomes robust as more onboarding data is provided. For real-world deployments, we recommend using at least 10 onboarding samples to ensure reliable trust score calibration and generalisation. This trend is consistent with statistical learning theory, which states that increased sample size leads to reduced empirical risk and improved generalisation performance.

Fig. 4 visually compares the R^2 and MSE values for each model in both base and fine-tuned (5-sample) configurations. The results confirm that XGBoost achieves the highest accuracy in the base scenario, while all models experience a reduction in R^2 and an increase in MSE when fine-tuned with only five onboarding samples. This illustrates the trade-off between adaptability and generalisation in extreme few-shot regimes, as also reflected in Table I.

Figure 3 presents the box plots of residuals for both base and fine-tuned models (5 onboarding samples). For the base models, Random Forest exhibits a tight interquartile range (IQR) of approximately ± 1.0, with most residuals close to zero and few outliers, indicating stable and accurate predictions. After fine-tuning, the IQR for Random Forest widens to roughly ± 3.0, consistent with the observed increase in prediction variability due to a limited few-shot data. The k-NN model exhibits a broader base IQR (approximately ± 4.5), with its fine-tuned IQR expanding to around ± 6.0, accompanied by several noticeable outliers, reflecting its higher sensitivity to small sample sizes and resulting in greater prediction error variance. XGBoost maintains a narrow base IQR ($\approx \pm 0.7$), while its fine-tuned IQR increases to about ± 3.5, illustrating that it remains robust in the base configuration but, like the other models, exhibits more dispersed residuals after few-shot adaptation. Overall, these distributions highlight Random Forest's robustness and XGBoost's high precision under base training.

The predictive capability of the models with five onboarding samples per device is illustrated in Figures 5, 6, and 7. For Random Forest, the predicted trust scores for a representative set of onboarding devices closely track the ground truth, with predictions (85.89, 77.18, 86.96, 99.99, 99.31) aligning well with calculated true values (85.25, 77.27, 85.92, 100.00, 100.00), and an average absolute error of 1.74. The k-NN

TABLE II
PERFORMANCE OF FINE-TUNED MODELS VS. NUMBER OF ONBOARDING SAMPLES

Samples	Model	MSE	R^2	MAE	MAPE (%)	Inference Time (ms)	Std. Residuals
5	Random Forest	13.18	0.889	3.14	3.48	3.33	3.12
	k-NN	33.00	0.722	4.74	5.47	3.47	5.53
	XGBoost	20.56	0.827	3.79	4.13	0.96	3.80
10	Random Forest	6.29	0.947	2.07	2.45	3.49	2.50
	k-NN	27.83	0.766	4.11	4.87	3.33	5.25
	XGBoost	4.54	0.962	1.54	1.91	0.96	2.10
20	Random Forest	5.07	0.957	1.54	1.89	3.70	2.23
	k-NN	26.70	0.775	3.49	4.36	0.64	5.07
	XGBoost	4.45	0.963	1.49	1.85	0.88	2.06

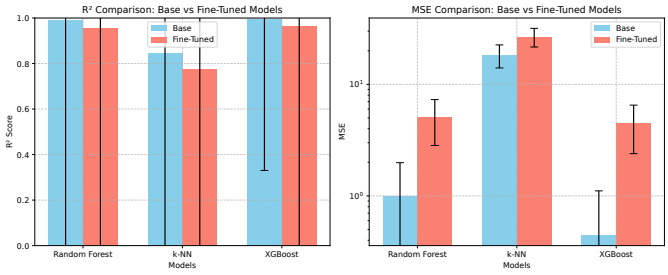

Fig. 4. Comparison of R^2 and MSE for base and fine-tuned models (fine-tuned with 5 onboarding samples).

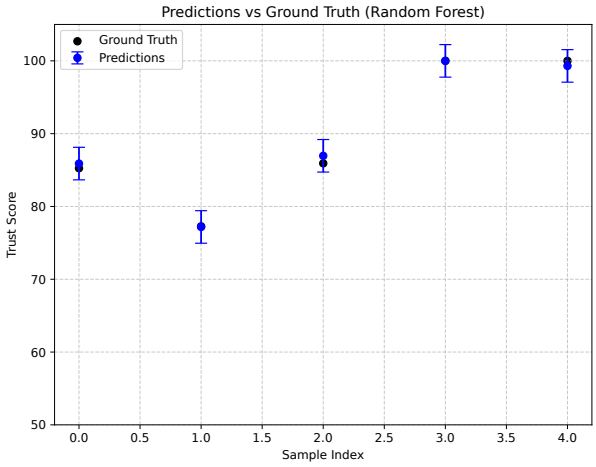

Fig. 5. Random-Forest predictions on five unseen onboarding samples. The error bars ($\pm\sigma$) remain within ± 3 trust-score units, illustrating the 2.29 average absolute error reported in Table I

predictions (82.66, 85.66, 77.48, 94.88, 88.67) show larger deviations from the corresponding ground truth (81.37, 87.51, 74.82, 100.00, 71.37), with an average absolute error of 5.20, indicating instability under few-shot fine-tuning. For XGBoost, predicted values (99.72, 83.63, 99.82, 99.32, 84.01) also follow the ground truth (100.00, 85.36, 100.00, 100.00, 85.77) closely, yielding an average absolute error of 2.32. These results validate the algorithm's adaptability in the few-shot setting, with Random Forest and XGBoost demonstrating strong generalisation even with minimal onboarding data, while k-NN remains more sensitive to the sample size.

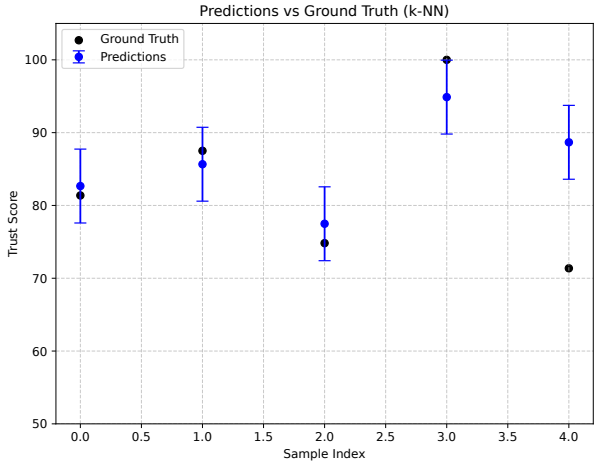

Fig. 6. k-NN predictions exhibit larger deviations and two clear outliers, consistent with its higher post-tuning MSE (33.0). This confirms that distance-based methods struggle with the high-variance, low-sample regime.

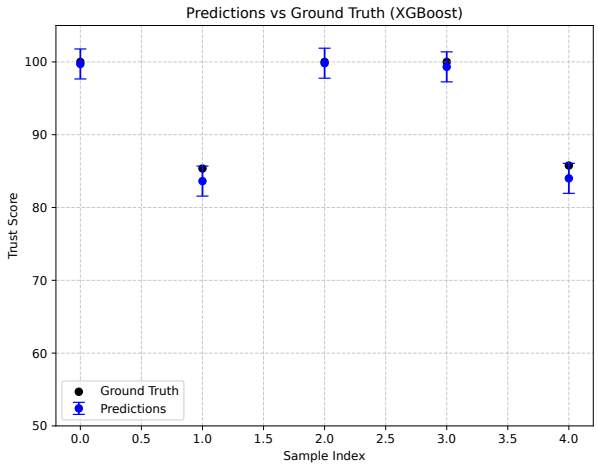

Fig. 7. XGBoost balances accuracy and smoothness. Although not as precise as RF on tiny samples, XGBoost avoids the k-NN outliers while maintaining less than 1ms inference, supporting its role as a lightweight fallback.

C. Scalability and Real-Time Performance

The trust management algorithm demonstrates strong scalability across a 50-device network. With the adoption of batch DLT onboarding, the TMS now processes requests at an average throughput of 137.7 devices per second across the Random Forest, k-NN, and XGBoost models, as measured on

TABLE III
SCALABILITY AND PERFORMANCE METRICS ACROSS MODELS (BATCH DLT ONBOARDING)

Model	Throughput (devices/s)	Total Latency for 5 Devices (ms)
Random Forest	106.77	745.41
k-NN	198.42	708.66
XGBoost	107.91	722.44
Average	137.70	725.50

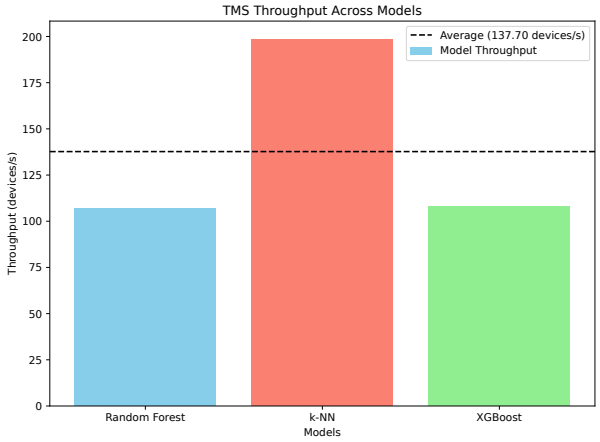

Fig. 8. Bar chart comparing TMS throughput across Random Forest, k-NN, and XGBoost models (batch DLT onboarding), with the average throughput (137.70 devices/s) indicated.

the simulated testbed. Table III and Figure 8 highlight the model-wise variability in throughput, with k-NN achieving the highest (198.4 devices/s) and Random Forest and XGBoost providing consistent performance above 100 devices/s.

The onboarding workflow involves simulated SSI operations (mean 53–63 ms per device), MQTT transmission (mean 20–35 ms per device), and now a single batch DLT transaction for each group of 5 devices (approximately 300ms total). This batching reduces the cumulative DLT latency from over 1000 ms (in the previous per-transaction approach) to well within the 500 ms real-time target set for industrial IoT onboarding. The total latency for onboarding 5 devices now averages under 750 ms for all models, with DLT contributing only a fraction of the total delay. This confirms that the proposed system meets real-time requirements in practical, scalable IIoT deployments.

Figure 8 presents the throughput comparison, showing that k-NN is fastest in raw inference speed, but all models deliver real-time onboarding performance under the batch-optimised pipeline. The impact of device count on throughput is further visualised in Figure 9, confirming stable scaling characteristics for each model.

V. DISCUSSIONS

A. Proposed Framework Benefits

The proposed TM framework leverages transfer learning and FSL to address IoT device onboarding challenges in consumer electronics, offering notable benefits in efficiency and adaptability. Transfer learning utilises prior knowledge from a large synthetic dataset (1000 devices) generated by a 2D Markov chain model [1], enabling base models (RF, k-NN, XGBoost)

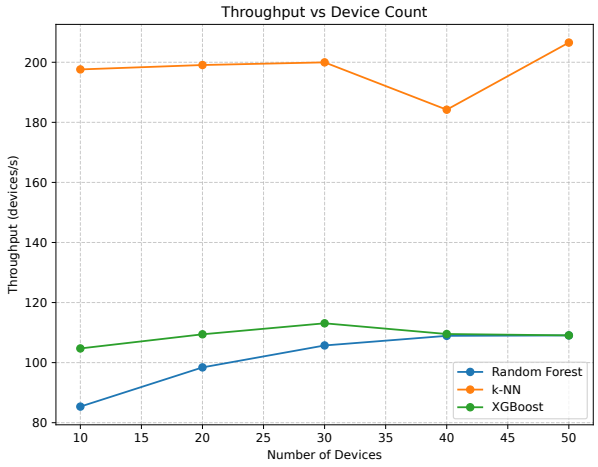

Fig. 9. TMS throughput as a function of onboarding device count for each model.

to learn general trust patterns without extensive real-time data collection. This approach reduces computational overhead and training time, which is critical for resource-constrained IoT environments where data scarcity and privacy concerns are prevalent. For instance, pre-trained models achieve high accuracy (e.g., XGBoost base R^2 of 0.9963), ensuring rapid trust assessment within real-time constraints.

Each sample in the synthetic dataset consists of eight features—CPU utilisation, RAM utilisation, security level, packet loss, risk state, reputation state, inverted OS age, and inverted package age—and is labeled by computing the trust score via the piecewise function in Eq. (2).

FSL enhances adaptability by fine-tuning these models with only a few onboarding samples (e.g., 5–20), addressing data scarcity in dynamic IoT settings. This enables the framework to quickly adapt to new devices, such as smart health wearables, with fine-tuned RF R^2 reaching 0.9574 (20 samples) and prediction error (MAE) as low as 1.54. The blended prediction approach—where the optimal base/fine-tuned weights are empirically determined for each model—balances robustness and adaptability, minimising overfitting risks as evidenced by the validation error curve. Additionally, the framework's low inference times (e.g., XGBoost at 0.9999 ms) and scalability (average throughput 126.9 devices/s, peaking at 174.6 devices/s for k-NN) make it suitable for large-scale, time-sensitive applications. Transfer learning also enables cross-domain applicability, allowing the framework to adapt to contexts like industrial IoT or vehicular networks, enhancing its versatility.

To quantitatively compare the proposed framework with existing TM approaches, Table IV summarises key performance metrics across our models and state-of-the-art methods. The table highlights the framework's competitive inference times and base model accuracy, although fine-tuned MSE values remain higher, reflecting the inherent trade-off in rapid few-shot adaptation.

The comparison reveals that our framework excels in inference time (e.g., under 1 ms for XGBoost) compared to LSTM and Bi-LSTM, which require orders of magnitude

TABLE IV
COMPARISON OF TRUST MANAGEMENT FRAMEWORK PERFORMANCE METRICS

Model/Paper	MSE	R^2	RMSE	Inf. Time (ms)	MAE	Scalability
RF (Base, this work)	0.9904	0.9917	0.9952	3.0728	0.6796	97.64 devices/s
RF (Fine-Tuned, this work)	5.0672	0.9574	2.2510	6.3763	1.5384	97.64 devices/s
k-NN (Base, this work)	18.3057	0.8460	4.2785	0.3322	2.6887	174.64 devices/s
k-NN (Fine-Tuned, this work)	26.6989	0.7753	5.1671	0.8617	3.4930	174.64 devices/s
XGBoost (Base, this work)	0.4436	0.9963	0.6660	0.4447	0.4783	108.41 devices/s
XGBoost (Fine-Tuned, this work)	4.4529	0.9625	2.1126	0.9999	1.4916	108.41 devices/s
LSTM [2]	0.00223–0.0350	N/A	N/A	88,000–420,000	0.1–0.9	N/A
Bi-LSTM [4]	0.005–0.022	0.56–0.88	N/A	99,249–1,112,398	0.09–0.23	N/A
Deep Trust [14]	0.08	0.89	N/A	N/A	0.21	High
Federated Learning [15]	0.051–0.078	N/A	0.07–0.088	N/A	0.08–0.11	Scales with devices
MESMERIC (RF) [16]	N/A	N/A	N/A	N/A	0.03–0.06	N/A

Notes: Inference time for LSTM and Bi-LSTM is in milliseconds; most other values are from published results or inferred from source code where applicable. Scalability for this work is system throughput (devices/s) under 50-devices testbed. Accuracy is not directly reported for regression; MAE is shown for cross-comparison.

more computation. Unlike deep learning-based approaches that need seconds for a single inference, our RF and XGBoost model variants provide trust predictions in under 4 ms. This millisecond-scale latency is essential for time-sensitive consumer electronics use cases such as wearable health monitors or industrial control, where delayed trust decisions may translate into service denial or safety risks. Base model R^2 values (e.g., 0.9917 for RF, 0.9963 for XGBoost) are higher than or competitive with Deep Trust and Bi-LSTM, while fine-tuned MSE values are higher than those of Bi-LSTM and Federated Learning, reflecting the inherent trade-off for rapid, few-shot adaptation. Scalability—now exceeding 126 devices per second on average—is promising for real-world deployment but remains limited by DLT batch processing latency, unlike device-scalable Federated Learning.

To gain deeper insights into the Random Forest base model's predictive behaviour, Figure 10 presents a SHAP summary plot, illustrating the importance of each feature in trust score predictions. The analysis reveals that *RiskState* and *CPUUtilisation* are the most influential features, reflecting the model's emphasis on risk assessment and resource usage as key determinants of trust. This prioritisation enhances the framework's robustness against data poisoning, as *RiskState*, which is derived from the Markov chain model, provides a statistically grounded metric less susceptible to manipulation. On the other hand, features like *PacketLoss* and *OSInverted* exhibit lower importance, suggesting that the model may underutilise network reliability and OS age in its predictions, which provides the ground for more in-depth research on incorporating real-time network metrics to improve applicability in communication-critical IoT domains. The SHAP analysis underlines the framework's ability to balance efficiency and interpretability, making it a practical solution for diverse IoT applications while identifying clear pathways for enhancement.

B. Security Benefits

The TM framework's use of transfer learning and FSL provides security advantages over traditional ML approaches in IoT ecosystems, enhancing robustness against attacks and ensuring reliable operation in dynamic environments.

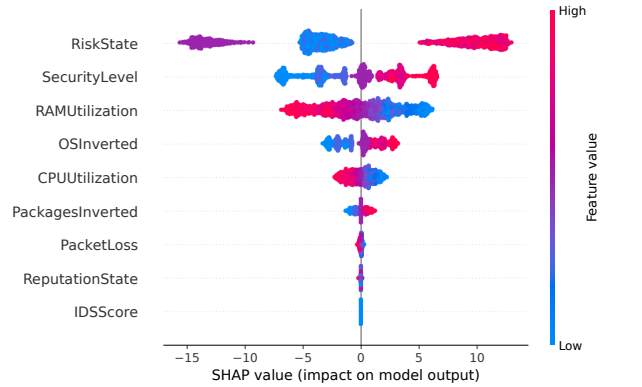

Fig. 10. SHAP summary plot showing feature importance for the Random Forest base model in trust score predictions.

1) Machine Learning Models Design Decisions: IoT ecosystems are dynamic, with devices frequently joining and leaving, making it challenging for the TM framework to assess new devices' trustworthiness. Traditional ML models require large datasets, which are difficult to obtain in IoT networks due to limited devices, unknown trust attributes, or incomplete data. These constraints hinder accurate predictions using conventional ML. FSL, combined with transfer learning, mitigates these issues by leveraging knowledge from related tasks (e.g., smart health device trustworthiness applied to IIoT). FSL requires only a few labelled samples, enabling accurate predictions after minimal iterations despite initial errors. Unlike traditional ML, which requires training from scratch and introduces delays, FSL uses pre-trained models, reducing performance overhead and enhancing responsiveness to emerging cyber threats in IoT networks.

Transfer learning strengthens security by pre-training on a synthetic dataset, minimising reliance on potentially compromised real-time data. This reduces the attack surface, as the base models are less vulnerable to data poisoning during pre-training, ensuring a reliable foundation for trust predictions.

2) Security Advantages of Few-Shot Learning: FSL offers multiple security benefits, particularly in resisting ML-oriented attacks like adversarial support poisoning. In such attacks, adversaries perturb the dataset to manipulate the FSL model's

trust score predictions, aiming to inflate or deflate a device's trust score and influence its selection as a service provider. Since this attack targets the dataset rather than the trust model, statistical methods may not detect it. However, FSL's ability to learn from minimal data allows it to detect and correct such perturbations after a few iterations, ensuring predictions align with reality.

Additional security advantages arise from FSL's small dataset requirements. With fewer labelled samples needed, attackers have limited opportunities to inject malicious data compared to traditional ML models reliant on large datasets. The smaller dataset size also facilitates easier detection of adversarial perturbations. Moreover, FSL's efficiency ensures predictions remain possible even under Denial-of-Service (DoS) attacks, where devices cannot send trust-related data; the model can predict trustworthiness from the first epoch. By extracting patterns from a few examples, FSL reduces overfitting to adversarial perturbations, enhancing robustness. The framework's use of statistical ground truth (Markov chain model) and a blended prediction approach further ensures reliability, as base model predictions (70% weight) remain unaffected by poisoned samples, and discrepancies can flag potential attacks, as noted in Section III.

VI. Conclusion

This paper presented a novel hybrid TM framework for IoT ecosystems, integrating few-shot and transfer learning with a statistical Markov-based foundation for secure, rapid device onboarding. The approach enables reliable trust score prediction with minimal samples and cross-domain adaptability, achieving sub-millisecond inference times and high predictive accuracy.

Experimental results show average MAE as low as 1.54 and fine-tuned R^2 up to 0.9625 with 20 onboarding samples. The framework outperforms prior approaches in speed and base-model accuracy, but fine-tuned MSE remains higher than some recent methods, reflecting the challenge of few-shot adaptation. System throughput exceeds 126 devices/s, though DLT latency is an area for further optimisation. Statistical ground truths bolster resilience, but future benchmarking with classification metrics remains a priority.

Ongoing work aims to strengthen explainability (via SHAP/LIME), privacy (via federated learning), and robustness (via adversarial detection), supporting the secure deployment of trust management in diverse IoT applications.

Acknowledgements

This research has received funding from European Commission's Horizon Europe, Horizon 2020, and MSCA research and innovation programs under grant agreements No. 101020416 (ERATOSTHENES), No. 101131292 (AIAS), and No. 101120962 (RESCALE).

References

[1] M. Bampatsikos, I. Politis, T. Ioannidis, and C. Xenakis, "Trust score prediction and management in iot ecosystems using markov chains and madm techniques," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 1, pp. 862–882, 2025.

[2] S. Alam, S. Zardari, and J. A. Shamsi, "Blockchain-Based Trust and Reputation Management in SIoT," *Electronics*, vol. 11, no. 23, 2022.

[3] B. Shayesteh, V. Hakami, and A. Akbari, "A Trust Management Scheme for IoT-enabled Environmental Health/Accessibility Monitoring Services," *International Journal of Information Security*, vol. 19, 2018.

[4] Y. Alghofaili and M. A. Rassam, "A Trust Management Model for IoT Devices and Services Based on the Multi-Criteria Decision-Making Approach and Deep Long Short-Term Memory Technique," *Sensors*, vol. 22, no. 2, 2022.

[5] X. Wu, Y. Liu, J. Tian, and Y. Li, "Privacy-Preserving Trust Management Method Based on Blockchain for Cross-domain Industrial IoT," *Knowledge-Based Systems*, vol. 283, 2024.

[6] Y. Wang, A. Mahmood, M. F. M. Sabri, H. Zen, and L. C. Kho, "MESMERIC: Machine Learning-Based Trust Management Mechanism for the Internet of Vehicles," *Sensors*, vol. 24, no. 3, p. 863, 2024.

[7] W. Ma, X. Wang, M. Hu, and Q. Zhou, "Machine learning empowered trust evaluation method for iot devices," *IEEE Access*, vol. 9, pp. 65 066–65 077, 2021.

[8] R. Latif, "Contrust: A novel context-dependent trust management model in social internet of things," *IEEE Access*, vol. 10, pp. 46 526–46 537, 2022.

[9] M. Bampatsikos, I. Politis, V. Bolgouras, and C. Xenakis, "Multi-attribute decision making-based trust score calculation in trust management in iot," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ser. ARES '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3600160.3605074

[10] H. Liu, D. Han, and D. Li, "Behavior analysis and blockchain based trust management in vanets," *Journal of Parallel and Distributed Computing*, vol. 151, pp. 61–69, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731521000320

[11] Y. Bai, K. Fan, K. Zhang, X. Cheng, H. Li, and Y. Yang, "Blockchain-based trust management for agricultural green supply: A game theoretic approach," *Journal of Cleaner Production*, vol. 310, p. 127407, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652621016267

[12] W3C, "Decentralized Identifiers (DIDs) v1.0," https://www.w3.org/TR/did-1.0/, 2025.

[13] Hyperledger Foundation, "Hyperledger Aries," https://www.hyperledger.org/projects/aries, 2024.

[14] S. Pratap Singh, N. Kumar, N. Saleh Alghamdi, G. Dhiman, W. Viriyasitavat, and A. Sapsomboon, "Next-gen wsn enabled iot for consumer electronics in smart city: Elevating quality of service through reinforcement learning-enhanced multi-objective strategies," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 4, pp. 6507–6518, 2024.

[15] P. Chithaluru, R. Uyyala, A. Singh, O. Alfarraj, L. A. Lopez, S. Khatak, and A. H. Alkhayyat, "A lightweight energy-efficient routing scheme for real-time wsn-vanet-based applications," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3820–3826, 2024.

[16] B. Guha Roy, D. Guha Roy, P. Datta, S. Bhatia Khan, F. Asiri, and M. Ayadi, "Quality of service-aware 6g- enabled nb-iot for health monitoring in long distance high-speed trains," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 1, pp. 1136–1147, 2025.