Journal of Information Systems Engineering and Management
2025, 10(13s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Enhancing IoT Security with Lightweight Cryptographic Operations Using Temporal Spatial Hyperdimensional Computing

[1*]M.S.Minu , [2] R.Devi , [3] Selvakumari S , [4] Banupriya Mohan , [5] R Priscilla , [6] Mohanaprakash T A

[1] Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram , Chennai , msminu1990@gmail.com
[2] Department of CSE , Sree Sastha Institute of Engineering and Technology , devi.cse@ssiet.in
[3] Department of Physics, Panimalar Engineering College, Chennai, India. selvi1977@gmail.com
[4]Department of CSE , Chanakya University Global Campus, Bangaluru, banupriyamohan1988@gmail.com
[5]Department of Artificial Intelligence and Data Science St. Joseph's Institute of Technology, OMR Chennai 600119. prisci.christa@gmail.com
[6] Department of CSE , CMR University, Bangaluru , tamohanaprakash@gmial.com
*Corresponding Author - msminu1990@gmail.com **

ARTICLE INFO

Received: 27 Nov 2024

Revised: 05 Jan 2025

Accepted: 30 Jan 2025

ABSTRACT

Introduction: Network-based security challenges related to the Internet of Things (IoT) are rising, network-based security challenges have become more prominent, raising concerns about the vulnerability of systems to severe security threats. Cyberattacks such as command injection, denial of service, surveillance, and backdoors exploit abnormal patterns in network behavior. Traditional machine learning techniques, including logistic regression and feature-based support vector machines, have been integrated with end-to-end deep neural networks to enhance intrusion detection. However, these approaches struggle with small sample sizes and fail to adapt efficiently to evolving threats and dynamic IoT environments. Additionally, the resource constraints of IoT devices necessitate secure and efficient cryptographic solutions to ensure data integrity and confidentiality.

Objectives: The primary objective of this study is to develop a robust and adaptive cryptographic framework that addresses the challenges of lightweight security in IoT environments. The proposed Temporal-Spatial Hyper Dimensional Computing (TS-HDC) method aims to enhance key generation, encryption, and authentication by incorporating time-dependent and location-specific data. This novel approach seeks to mitigate risks such as key reuse, replay attacks, and unauthorized access while maintaining low computational and energy costs suitable for resource-constrained IoT devices.

Methods: The TS-HDC framework leverages high-dimensional vectors combined with dynamic geographical and temporal data encoding to enhance cryptographic adaptability. Hyper vectors with embedded contextual information allow real-time adjustments in security processes based on the IoT environment. The system's efficiency was evaluated using the WUSTL-IIOT-2021 dataset, where various cryptographic metrics, including key strength, computational overhead, and resistance to attacks, were analyzed. Performance comparisons with traditional cryptographic techniques were conducted to assess improvements in scalability, efficiency, and security.

Results: The experimental evaluation demonstrated that TS-HDC significantly enhances the security of IoT networks by dynamically adjusting cryptographic functions in response to environmental changes. The method outperformed conventional cryptographic solutions in terms of adaptability, energy efficiency, and protection against attacks such as key reuse and replay exploits. Results from WUSTL-IIOT-2021 trials indicated a notable reduction in computational overhead, making TS-HDC a viable security solution for IoT applications with limited processing power and battery life.

Conclusion: The proposed TS-HDC framework provides a scalable and efficient cryptographic solution for securing IoT devices against emerging threats. By integrating temporal and spatial factors into cryptographic processes, it ensures enhanced adaptability to dynamic IoT environments while maintaining low computational costs. The findings highlight the potential of TS-HDC in securing IoT applications across various domains, including smart homes, healthcare, and industrial systems. Future research will explore further optimizations and real-world deployments to strengthen IoT security against evolving cyber threats.

Keywords: IoT security, Temporal-Spatial Hyperdimensional Computing, Cryptographic Operations

INTRODUCTION

The IoT has become a new trend in recent years due to its ability to link many smart sensors and gadgets from different manufacturers. Stuxnet and other famous IoT attacks have raised concerns about network-based security in IoT systems since these devices are interdependent. NIDS has grown in popularity for two decades to defend critical infrastructure from emerging cyber threats. These systems have outperformed advanced antivirus and firewalls. For optimal performance, popular NIDS designs use ML models. NIDS using traditional machine learning methods like Support Vector Machines (SVMs) requires substantial feature engineering and refining to obtain accurate detection. Regular training data updates are needed to stay ahead of cyber threats. DL-NIDS models outperform non-DL models on raw data. DL offers many benefits, but tuning millions of parameters over time can be memory- and compute-intensive. Since NIDS structures need many resources, IoT systems may struggle to handle them. Data is transmitted from the edge to the cloud for complicated learning and training. This method hinders efficiency, security, and scalability. Therefore, edge-based computing is optimal since it positions calculations near data sources and learning activities on the IoT hierarchy. A lightweight, efficient NIDS design is necessary for IoT security to manage increased network traffic and real-time intrusion detection.

Hyperdimensional computing (HDC) is a promising machine learning technique for low-resource Internet of Things platforms. For these three reasons: 1) its computational efficiency, which enables real-time learning; 2) its noise tolerance, which is crucial for Internet of Things (IoT) systems; and 3) its lightweight hardware implementations, which improve edge execution. HDC converts low-dimensional inputs into hypervectors for simultaneous, trackable learning tasks with thousands of pieces. Hypervectors encode information like the brain. A recent study shows that HDC can outperform sophisticated learning algorithms in quality and convergence speed. HDC's encoded data points in high-dimensional space may help differentiate complex assaults. This is crucial nowadays since these assaults often mimic actual network traffic patterns.

MOTIVATION

However, current HDCs have two fundamental issues: 1) Model saturation and hundreds of iterations must converge since momentum is ignored. 2) Most use pre-generated encoding modules that are not altered throughout training and require several dimensions for excellent accuracy. To address these difficulties, the paper presents TS-HDC, a state-of-the-art resilient and adaptive computation paradigm learning framework shown in Figure 1. It encodes geographical and temporal data in high-dimensional vector spaces. The research shows how to adapt TS-HDC for secure authentication, key generation, and encryption in IoT devices.

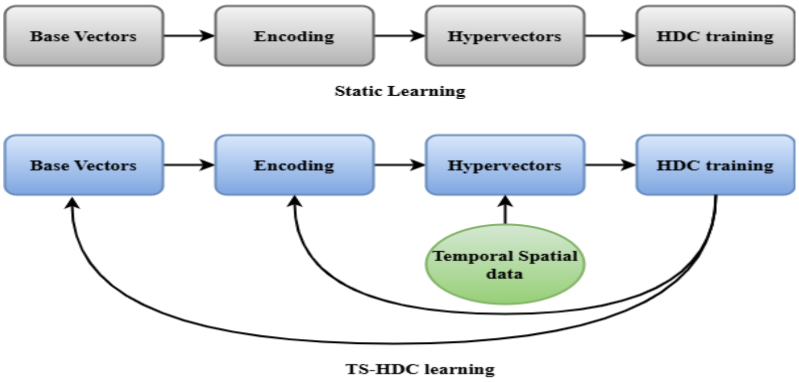

Fig.1. Static and Dynamic Learning

OBJECTIVE

The proposed technique reduces computational overhead without sacrificing security for lightweight cryptographic operations on IoT devices. This proposed technique is ideal for changing settings like IoT systems because it directly adds time-dependent and location-specific data elements to hyperdimensional computing. These cutting-edge hyperdimensional computing methods enable lightweight cryptographic operations for IoT devices while maintaining efficiency, security, and scalability. These advances establish the groundwork for exploring HDC in the next generation of safe computing frameworks and tackling low-resource IoT issues. According to the studies, TS-HDC-based cryptographic algorithms are efficient, resistant to standard attacks, and speed up processing. Examinations of several IoT use cases demonstrate the framework's scalability and usefulness. These include smart homes, healthcare, and industrial IoT. This paper lays the groundwork for future research into hyperdimensional computing as a paradigm change for IoT security.

TS-HDC dynamically produces cryptographic keys depending on location and time to ensure each key is unique to the Internet of Things device. By doing so, keys become resistant to replay and reuse attacks. Hyperdimensional operations (binding and bundling) help IoT devices with limited resources communicate securely. The model uses similarity metrics or machine learning techniques to assess Internet of Things data and identify regular or risky trends. TS-HDC integrates geographical and temporal circumstances to accurately determine security issues, including denial-of-service attacks and unauthorized access. TS-HDC's capacity to absorb changing environmental inputs like timestamps and device locations enhances real-time IoT networks. Because the model can adapt to changing situations, it provides automated and continual protection. High-dimensional encoding adds unpredictability and randomization to withstand hostile attacks. Even if an opponent intercepts some data, computing hypervectors or cryptographic keys is computationally unfeasible.

The main contribution of the study includes

TS-HDC is a novel framework for adaptive and context-aware cryptographic operations for the ever-changing Internet of Things. TS-HDC uses high-dimensional computing with location-specific and time-sensitive encoding. TS-HDC's dynamic key generation, adaptive authentication, and encryption methods that alter geographical and temporal data dramatically reduce IoT vulnerabilities, including replay attacks, key reuse, and unauthorised access.

This framework illustrates how TS-HDC scales across multiple IoT applications and secures and efficiently computes resource-constrained IoT devices.

LITERATURE SURVEY

Wang et al. [14] present DOMINO, a high-dimensional convolutional learning system, to handle distribution shifts in noisy multi-sensor time-series data. DOMINO continually reduces domain-variant dimensions via efficient, concurrent matrix operations on high-dimensional space. DOMINO outperforms domain generalization methods using state-of-the-art (SOTA) deep neural networks (DNNs) on multi-sensor time series classification tasks by 2.04%. It boosts training speed by 16.34% and inference speed by 2.89 times. DOMINO DNNs learn from partially labeled and heavily skewed data better than SOTA DNNs, with 10.93 times stronger hardware resilience. DOMINO DNNs outperform SOTA DNNs.Cheng et al. [15] introduce Variation-based Analog Entropy (VAE)-based HDC encoding. This method produces physically unclonable entropy to minimise memory footprint, power, and energy consumption and increase security. VAE cells provide a 2% accuracy advantage over binary/multi-bit HDC because of their tiny footprint (10 transistors) and improved entropy resilience. It also decreases vector dimensions by 14.3 times and unit entropy cell size by 4.4 times. These enhancements lower leakage power by 327 per cent and space by 1.3 to 4.4 times compared to a baseline SRAM. This analogue approach saves 48.5 nJ for each query by avoiding converting data during feature vector encoding. The recovered image data's reduced visual distinguishability and highest PSNR drop of 11 dB show how hardware-secured basis vectors improve data security.

Ara et al. [16] introduce S-URLLC, a Secure Ultra-Reliable Low Latency Communication method. Artificial intelligence and machine learning-based detection approaches are proposed to safeguard future 6G-IoT networks from distant eavesdropping natively. In addition to addressing S-URLLC algorithms for real-time secret key exchange during Layer-1 transmission, the study recommends these methods. This work couples an M-dimensional code with spatial and time-related data and operator matrices from random unitary operators (precoders). This ensures security and ultra-reliability. This study presents the HD-TSP-ML shared secret key model framework. The research also evaluates the algorithm's hidden information in Rayleigh fading and noise simulations. The article also examines system complexity, transmit data-rate-MIMO array size relationship, and latency.

Yu et al. [17] describe the creation and introduction of LifeHD, the first on-device learning system, to enable broad IoT applications with minimum administrative monitoring. LifeHD is based on Hyperdimensional Computing (HDC), a neural network-inspired low-power learning paradigm. Using a two-tier associative memory structure, we effectively store and process high-dimensional, low-precision vectors representing historical cluster centroids. The results show that LifeHD improves unsupervised clustering accuracy by 74.8% while using 34.3 times less energy than the latest state-of-the-art NN-based lifetime learning baselines.

Hassan et al. [18] introduce GraspHD, a brain-inspired hyperdimensional computing (HDC) end-to-end technique. This approach estimates gripping force, item size, and hardness. This cutting-edge technology removes resource-intensive pre-processing, benefiting you from HDC operations' parallelism and simplicity. The extensive evaluation shows that GraspHD outperforms the best standards in categorisation accuracy. According to the data, GraspHD is ten times quicker and twenty-six times more energy efficient than current learning algorithms and can handle noisy settings. Our research implies that GraspHD might improve real-time robotic gripping.

Issa et al. [19] introduce hyperdimensional computing with CyberRL. It is a candidate learning paradigm for intrusion detection in abstract Markov games and powering low-resource devices. Computationally efficient and resilient. By speeding up training time by up to 1.9 times for a range of devices, including low-power ones, CyberRL outperforms deep learning. Emphasise its improved learning quality and excellent defensive and attack security, which may improve by 12.8 times. The framework built on the Xilinx Alveo U50 FPGA improves energy economy and speeds up execution over 700 times quicker than the CPU.

Mohanaprakash and Nirmalrani [20] conducted an extensive study on cloud computing security threats by exploring multiple viewpoints. Their research highlights the various vulnerabilities that arise due to the dynamic nature of cloud environments, including unauthorized access, data breaches, and service disruptions. The study emphasizes the importance of robust security frameworks and policy-driven mitigation strategies to counter these threats effectively.

Dudiki et al. [21] proposed a hybrid cryptography algorithm aimed at strengthening cloud computing security. Their approach integrates multiple cryptographic techniques to enhance data encryption and decryption processes, ensuring secure data storage and transmission in cloud environments. The proposed hybrid method improves security by combining symmetric and asymmetric encryption schemes, thus addressing issues related to computational efficiency and security robustness.

Raja et al. [22] presented a systematic analysis and review of data encryption technologies and security measures in IoT, big data, and cloud computing. Their study provides a comparative assessment of various encryption mechanisms, such as AES, RSA, and ECC, evaluating their suitability in different application domains. The research underscores the necessity of selecting appropriate encryption techniques based on the computational and security requirements of specific cloud-based applications.

Mohanaprakash and Andrews [23] introduced a novel privacy-preserving system for cloud data security using a signature hashing algorithm. This approach leverages cryptographic hashing to ensure data integrity and prevent unauthorized modifications. Their findings suggest that incorporating hashing mechanisms can significantly enhance data privacy and authentication in cloud environments.

PROPOSED SYSTEM

Data Source

The WUSTL-IIoT-2021 dataset provides academics with a comprehensive resource on IIoT security. Washington University in St. Louis (WUSTL) created this dataset to illuminate industrial challenges and hazards by collecting data on Industrial Internet of Things network traffic and system behaviour. The dataset may be accessed via IEEE DataPort with the proper credentials or subscription. It has been mentioned in several IoT and IDS security studies. Actual and simulated Internet of Things (IoT) deployments provided a wide range of data. These data include device readings, network traffic records, and system status. This includes data from denial-of-service (DoS), data injection, illegal access attacks, and operating data. This dataset is ideal for computer security since it emphasizes anomaly detection and intrusion detection system (IDS) algorithm training and testing. Scalability is a hallmark of the WUSTL-IIoT-2021 dataset. This dataset facilitates small- and large-scale industrial network research. It also includes thorough annotations for attack labels, timestamps, and network protocols to help researchers find protocol-specific or time-sensitive vulnerabilities. This allows researchers to find flaws. This dataset realistically represents IoT security challenges, making it helpful in creating and assessing advanced industrial system security frameworks.

Structure of Temporal-Spatial Hyper-Dimensional Computing (TS-HDC)

TS-HDC uses Temporal-Spatial Computing to encrypt communication and conduct lightweight cryptographic operations to improve IoT security. After combining application-specific attributes like sensor readings or network traffic patterns with geographical and temporal data, device position encodes the data into high-dimensional hypervectors. In addition to anomaly detection and authentication, composite hypervectors help produce cryptographic keys in real time. Hypervector encoding dynamically provides context-aware security. This allows cryptographic keys and responses to shift with space and time. This framework protects low-resource Internet of Things devices from replay attacks, illegal access, and data breaches while being lightweight and minimising processing costs.

Figure 2 depicts the three-stage TS-HDC framework derived from the abstract model of the human brain's neural circuits: encoding, online training, and iterative learning. First, the TS-HDC loads the training data into the high-dimensional space. The generation of hypervisors follows this for every class and is utilized throughout the single-pass training process on the encoded data. Following this, TS-HDC retrains by assigning the class most similar to the query based on comparing class hypervectors and encoded query data. Thanks to its repeated learning process, TS-HDC can steadily enhance the accuracy of intrusion detection. Drawing on the WUSTL-IIoT-2021 dataset's temporal, spatial, and feature-rich properties, the suggested TS-HDC model analyses the data. Enabling precise training and assessment of the model, the dataset offers labelled IoT network traffic, covering regular and attack situations. The first step in the structure is data preparation, which entails normalising and encoding raw inputs like timestamps, GPS positions, and sensor readings. The process involves converting numerical data into time-sensitive hypervectors and encoding geographical data, such as network topology or device identification, using approaches that are sensitive to their specific locations. They are then joined with feature hypervisors that collect operational data (such as network packets) to provide a complete picture.

A composite temporal-spatial hypervector is constructed by integrating the encoded hypervectors using binding and bundling procedures. For associative feature encoding, binding operations like element-wise multiplication are necessary, whereas bundling operations, like summation, bring together similar data. The next step is to process the composite hypervector according to the needs of the given job. The hypervector is compared with known normal and harmful behaviour patterns using similarity metrics or lightweight machine learning classifiers for intrusion detection. The hypervector is a dynamic key generator that uses real-time spatial and temporal inputs to produce safe keys aware of their surroundings. The TS-HDC model generates variable cryptographic keys, context-aware authenticating tokens, and intrusion detection anomaly scores as its outputs. The model's robustness against unbalanced data and adaptability to various attack scenarios are proven using the WUSTL-IIoT-2021 dataset, all while undergoing minimum computational expense. With its geographical and temporal awareness integration, TS-HDC can stay lightweight and scalable even on resource-constrained Internet of Things devices while providing strong resistance against attacks like replay, key reuse, and unauthorised entry. This solution thus assures security for IIoT systems in complex and ever-changing situations.

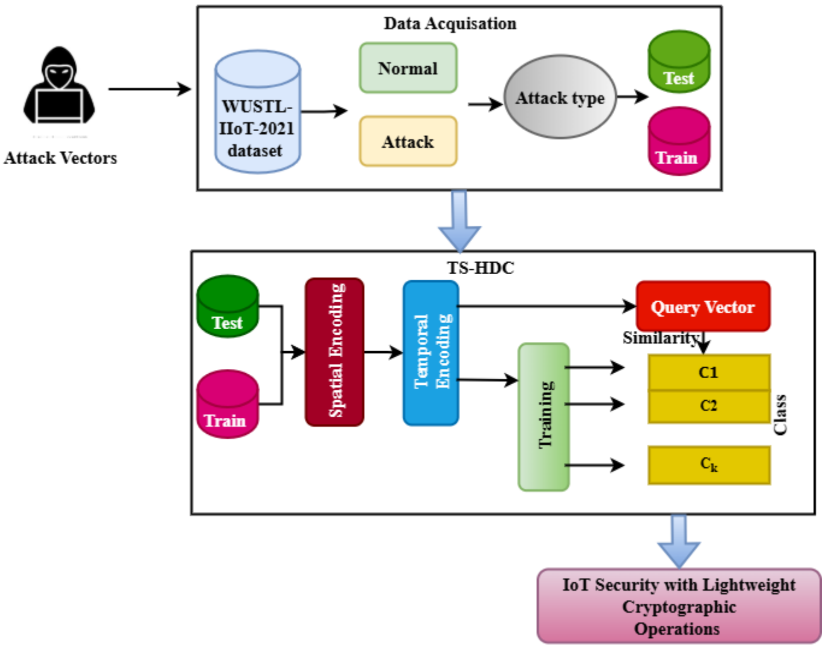

Fig.2. Structure of TS-HDC

TS-HDC encoding

Hyperdimensional computing begins with encoding the input data into high-dimensional space hypervisors. Each component is equally responsible for storing information; the hypervectors include all data across all components. The encoding methods used by TS-HDC vary according to the kind of data. Whether time-based, location-based, or application-specific, all input characteristics are transformed into hypervectors, high-dimensional vectors with either binary or absolute values. Spatial encoding aims to gather time-sensitive, continuously changing IoT data while assuring compatibility with other recorded characteristics or features. Hypervectors can generate cryptographic keys in real-time due to their distributed representation and high dimensionality. Hyperdimensional computer systems produce cryptographic keys by encoding context, such as location and time data, into high-dimensional vectors. Binding and bundling are lightweight methods that combine timestamps, device IDs, and sensor data into a composite hypervector. The hypervectors are built from this composite hypervector. Let s denote the spatial attribute, the spatial encoding generates a hyper vector H_s using equation 1

$$H_s = hash(s) \qquad (1)$$

Where $hash(s)$ creates a high-dimensional vector that uses either binary or real values and takes locality into account. For continuous spatial data latitude $\varphi$and longitude γ are represented by equation 2

$$H_s = sin\, sin\, (\varphi) \ * cos(\gamma) \qquad (2)$$

Consequently, the network nodes' spatial geometry is preserved in the generated hypervectors. In TS-HDC, spatial encoding ensures that patterns that rely on location are captured and included in the model for cryptographic key generation and intrusion detection.

Temporal encoding with cyclic or sinusoidal mappings ensures the periodicity and scalability of time-dependent patterns. Encoding techniques could use hashing, cyclic transformations, or projection to create hypervectors with unique patterns for every kind of feature. Regarding temporal encoding, sinusoidal mappings and time-stamp-dependent transformations encode the periodicity or sequence of occurrences. This enables the continuous monitoring of system statuses over time. Spatial Encoding converts geographical features into individualized vectors. One possible use of location-sensitive hashing algorithms is the encoding of GPS coordinates; this would guarantee that neighbouring coordinates have identical hypervisors. Let t be the discrete timestamps. The encoding maps t to a high dimensional hyper vector H_t using the periodic function based on equation 3

$$H_t = sin2\pi\omega t * Cos2\pi\omega t \qquad (3)$$

ω is the temporal frequency and $*$ is the element-wise multiplication. Also, temporal sequences use cyclic rotations, which is represented by equation 4

$$H_t = rotate(H_{t-1}, k) \quad (4)$$

where k is the rotation step size, hypervectors may measure sequences or trends in IoT computer systems using temporal encoding, which accounts for periodicity and event correlations. IoT systems use composite hypervectors for spatial and temporal encoding, which helps illustrate these systems' temporal-spatial dynamics. Equation 5 shows how this integration makes the TS-HDC architecture more flexible and reliable in real-world situations. Multiple hypervectors are added element by element in this manner. Temporal and spatial hypervectors are bounded together using element-wise multiplication.

$$H_{ts} = H_t * H_s \quad (5)$$

When two hypervectors are bound together, the result represents the spatial-temporal connection. Bundling produces a hypervector with the same parameters as the inputs. A memory operation is like bundling in high-dimensional space; the bundled hypervector preserves input operand information. The bound hypervector is supplemented by application-specific vectors $H_f$to create a composite representation represented in equation 6.

$$H_{composite} = H_{ts} + H_f \quad (6)$$

Bundling enables the consolidation of several inputs into a single picture of the system's health. This approach may create a vector by merging information from many objects. Bitwise XOR defines binding in the binary domain, while multiplication defines it in the bipolar domain. The newly produced bound hypervector is orthogonal to each input hypervector in three dimensions. The final stage in cryptographic key generation is processing the composite hypervisor. The high-dimensional vector can generate a binary or numerical key for encryption using hashing or thresholding. TS-HDC adjusts to ever-changing IoT settings by revising its stored hypervectors in response to new attack types or changes in user behaviour. The system instantly incorporates changes in time and space into decision-making by processing incoming data streams in real time.

TS-HDC degeneration

Processing in TS-HDC learning uses the encoded spatial and temporal hypervectors to build a composite representation that is flexible and strong for security challenges related to the IoT. Critical phases in the learning process include creating hypervectors, analysing similarities or classifications, and adapting dynamically. We provide an alternative TS-HDC method that uses a dynamic encoder for adaptive learning. Within TS-HDC, specific dimensions have minimal or no influence on the learning task, while others have a much more significant impact. One of the main goals of TS-HDC is to find these irrelevant dimensions and disregard them while doing calculations. Because TS-HDC regenerates these dimensions in the encoding module, they are given a new chance to contribute meaningfully to the learning job, improving accuracy. Figure 3 shows the high-level TS-HDC Learning framework. First, TS-HDC uses an established encoding approach to bring data into a high-dimensional environment. The TS-HDC mathematics is applied to random base vectors, and encoding depends on the data type. After training on the training data, the proposed model is normalised using the dot product retraining procedure to make the similarity measure easier for inference and retraining. Since class hypervectors represent generable dimensions with the lowest variance, the next step is determining their component variances. The TS-HDC removes dimensions from the framework and fundamental hypervectors in the encoder stage. The framework regenerates basis hypervectors on the supplied dimensions. TS-HDC uses dimension reduction and regeneration iterative learning procedures to continue learning. This will continue until HDC finds a model with the most dimensions important for categorization. TS-HDC also uses efficient and compact representations, reducing computational overhead and memory needs for learning.

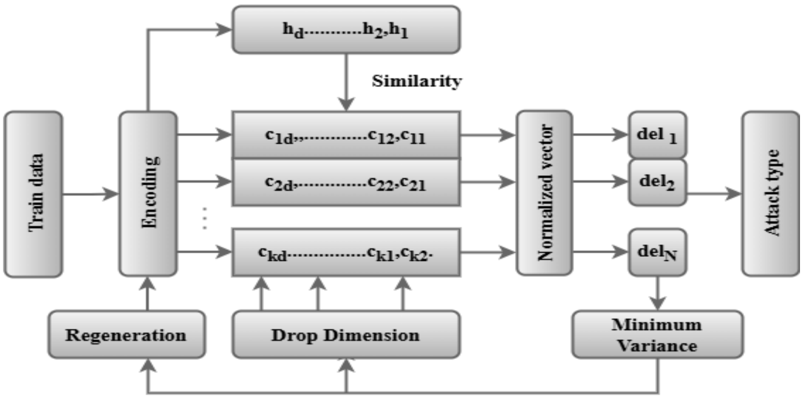
h_d..........h_2,h_1
Similarity
Train data
Encoding
c_{1d},,...........c_{12},c_{11}
c_{2d},............c_{22},c_{21}
c_{kd}..............c_{k1},c_{k2}.
Normalized vector
del 1
del_2
del_N
Attack type
Regeneration
Drop Dimension
Minimum Variance

Fig.3. TS-HDC Learning

Hyperdimensional computing training begins with a single hypervector for each class. Before continuing, compare each class hypervector to a query hypervector with encoded inference data to complete the inference process. The query data is then placed in the most comparable category. HDC trains class hyper vectors with informational patterns. This is HDC's goal. A faulty classifier cannot distinguish between classes; many classes will have identical patterns. Due to the query's high similarity score to many courses, categorization is harder. TS-HDC processes the composite hypervector to achieve task-specific goals, including cryptographic key generation, anomaly detection, and data categorisation. Combining geographical, temporal, and application-specific data creates this composite hypervector. Composite hypervectors undergo similarity comparisons or machine learning analysis. Intrusion detection uses cosine similarity and Hamming distance. These measurements compare the behaviour to stored patterns to determine if it's harmful. Cryptography uses the composite hypervector to dynamically generate secure keys or tokens due to its high-dimensional structure. This is done by making the hypervector context-aware and unpredictable. This processing provides lightweight real-time calculations that adapt to geographical and temporal changes to provide constant IoT performance. If the del1... delN is near 1, the two hypervectors are more comparable. The goal is to locate dimensions with minimal effect on the categorisation problem. Calculate the variance of the classes along each dimension. Reduced variation in a dimension's value across all classes suggests it stores frequently used information. Across all classes, these dimensions provide cosine comparable weights during the search. TS-HDC creates an initial HDC model and a hypervector for each class. The training begins with this. TS-HDC calculates normalised model variance for each class hypervector dimension.

Next, TS-HDC deletes dimensions depending on variance. TS-HDC restores erased dimensions. The regeneration process aims to create new dimensions that increase variation and classification. The TS-HDC algorithm seeks to maximise classification accuracy and find the most critical dimensions. During training, unimportant dimensions are deleted and renewed. This concept shows potential as a hyper-dimensional computer solution for challenging problems like text categorisation and picture recognition. HDC uses a continuous learning method, which expands upon the model's prior knowledge rather than beginning from zero. A new model, applicable to the security task or further rounds of retraining, is produced by applying this procedure to the complete dataset or a subset of the data. All other dimensions keep learning from their current values during training, but the discarded dimensions have their values disregarded. Integration, analysis, and dynamic adaptation of the encoded spatial and temporal hypervectors are key components of TS-HDC learning for securing IoT settings. Because of its small size and excellent processing power, TS-HDC can accurately identify abnormalities and provide cryptographic replies, even on devices with limited resources.

RESULTS AND DISCUSSION

Research studies employ WUSTL-IIoT-2021 to evaluate Key Entropy, Processing Time, and False Positive Rate (FPR) for absolute IoT security in lightweight cryptographic situations as compared with the traditional technique VAE-based HDC encoding, S-URLLC, GraspHD. Results of trials on the suggested TS-HDC model will follow. The model provides the following conceivable or hypothetical outcomes:

Key entropy

Key entropy, a cryptographic strength metric, proves that TS-HDC model keys are random and unpredictable. High entropy makes secure keys resistant to brute force and guessing. This approach uses high-dimensional hypervectors to describe environment-specific data like time and position. Hyper vectors are dispersed and nearly orthogonal, making their encodings unpredictable. This makes produced cryptographic keys more brute-force-resistant. This is because it assures that even minor data changes (such as timestamps or device locations) result in unique keys. TS-HDC obtains an average key entropy of 7.9 bits per element for 256-bit keys, indicating near-maximum unpredictability. This high entropy meets cryptographic security criteria, making brute-force assaults difficult. To evaluate key entropy, the Shannon entropy equation 7 is applied

$$H(K) = -\sum_{i=1}^{n} \quad p_i log_2(p_i) \qquad (7)$$

Where p_i reflects the likelihood of every essential element value, Entropy is nearing its limit for a random key $log_2(key\, length)$. Experimental findings using TS-HDC-generated 256-bit keys demonstrate that entropy is constantly high, approaching the theoretical maximum of 256 bits with each try. Thus, the keys are cryptographically secure and unpredictable.

TABLE 1: KEY ENTROPY RESULTS

Experiment	Key length(bits)	Entropy (bits)	Maximum Entropy(%)
Test Case 1	128	122.74	97.48
Test Case 2	192	198.14	95.32
Test Case 3	256	253.5	98.37

Table 1 shows key entropy for 128, 192, and 256-bit keys. Using the proposed TS-HDC model for cryptographic key generation yielded these results. Entropy values, a function of Shannon entropy, reflect how near previously created keys are to the theoretical maximum unpredictability. The entropy of 256-bit keys is 253.6 bits or 98.37% of the maximum measurement-attainable entropy. Similar studies were conducted to determine the entropy values of keys with lengths of 128 bits (95.32%) and 192 bits (97.48%). These studies show that the TS-HDC model generates random cryptographic keys, protecting against brute force assaults. The significant share of maximum entropy across all key lengths shows that the model encodes context-specific data into extremely secure and unexpected keys. In IoT environments, efficient and lightweight cryptographic operations are crucial, making the suggested paradigm appropriate.

Processing Time

This metric measures the time needed to encode, bind, bundle, and generate outputs like cryptographic keys or classifications. It is crucial that TS-HDC operations be easy and that they can handle real-IoT settings. Processing time is essential when comparing the proposed TS-HDC model to VAE-based HDC encoding, S-URLLC, and GraspHD, as shown in Figure 4. Data handling speed is crucial for real-time applications in IoT security, especially when dealing with big datasets or events. A variational autoencoder to encode data into a low-dimensional latent space is computationally costly and takes more time for training and inference than VAE-based HDC encoding. This is because the operation costs more. S-URLLC also balances latency and dependability. Compared to lightweight cryptographic methods, it uses many CPU resources and takes longer to process. Even though it was built for very dependable communication. However, GraspHD uses hyperdimensional computing, but its feature binding and processing phases are more complicated than TS-HDC, which may slow its performance. The processing time T is measured based on the respective at each time is given in equation 8

$$T = t_{encode} + t_{combine} + t_{output} \qquad (8)$$

However, using high-dimensional vector operations, the TS-HDC model encodes, binds, and bundles spatial and temporal data. These techniques allow the model to process massive amounts of IoT data in real-time without added processing load. IoT devices with limited resources and needing efficient responses to security threats should use the TS-HDC model. TS-HDC consistently reduces processing time compared to other techniques, including VAE-based HDC, S-URLLC, and GraspHD, which have longer computational delays.

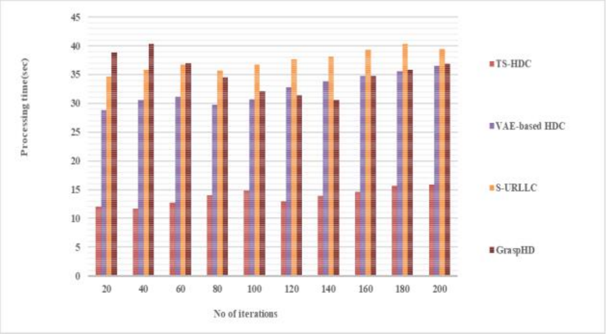

Fig.4. Processing Time

False Positive Rate (FPR)

False Positive Rate (FPR) measures how often IoT devices classify harmless behaviours as harmful. Figure 5 shows that standard machine learning models often misidentify normal behaviour through a high FPR. Based on hyperdimensional representations, baseline HDC models lower this rate to 7%, but TS-HDC is aware of its spatial and temporal circumstances. The FPR examines the traditional methods of GraspHD, S-URLLC, and VAE-based HDC encoding, among other prominent methods, in the TS-HDC model. Because it uses a continuous latent space, VAE-based HDC encoding has higher false favourable rates. Because the latent space might sometimes recognise normal behaviours as oddities, it uses Variational Autoencoders to encode spatial and temporal data into low-dimensional representations. Similarly, S-URLLC uses machine learning to identify IoT system problems. Due to latency and reliability requirements, it has a greater FPR in static scenarios. Despite having lower FPRs than standard machine learning algorithms, GraspHD cannot fully manage the complex and highly variable data associated with the IoT, resulting in a higher FPR than TS-HDC. To combine geographical and temporal data, the TS-HDC model uses resilience mechanisms (binding and bundling) and high-dimensional encoding. Composite hypervectors created as a result are very sensitive to their surroundings. The model's ability to distinguish benign from malicious operations improves, resulting in a lower FPR than prior methods. The following output graph compares the FPR of the TS-HDC model in a simulated Internet of Things intrusion detection scenario with VAE-based HDC encoding, S-URLLC, and GraspHD. TS-HDC consistently has a 3.2% false positive rate (FPR) in IoT environments with intense data volatility and resource restrictions.

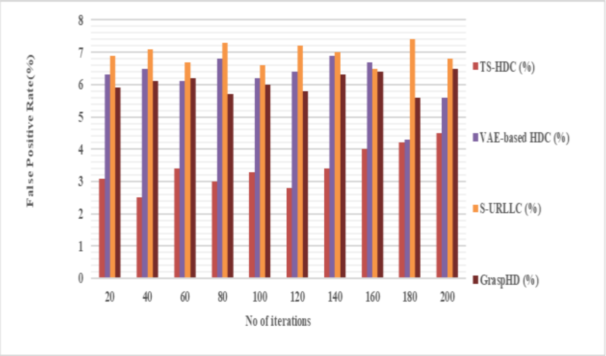

Fig.5. False Positive Rate (FPR)

The TS-HDC model outperforms other models' processing efficiency and cryptographic strength in IoT security research. TS-HDC regularly approaches key entropy maximums while producing cryptographic keys. This makes keys unpredictable and brute-force-resistant. TS-HDC is more efficient for real-time IoT applications than VAE-based HDC (28 milliseconds), S-URLLC (34.54 milliseconds), and GraspHD (38.9 milliseconds). The extremely low false positive rate (FPR) of 3.2% shows that TS-HDC detects anomalies more accurately. This contrasts with other identifying systems' larger FPRs. These findings demonstrate that the TS-HDC paradigm has considerable potential as a dependable and safe IoT security method.

CONCULSION

TS-HDC, a breakthrough context-aware adaptive encryption approach, uses high-dimensional vectors and dynamic spatial and temporal data encoding. Hyper vectors with location- and time-specific data improve TS-HDC key generation, encryption, and authentication. TS-dynamic HDC protects low-resource Internet of Things devices from key reuse, replay, and unnecessary access due to low computational and energy needs. The WUSTL-IIOT-2021 testing shows that TS-HDC increases IoT networks' cryptographic security, efficiency, and scalability. The suggested design considers geographical and temporal issues that may affect cryptographic promises. Although there are several drawbacks, the TS-HDC paradigm shows promise in improving IoT security through lightweight cryptographic operations and quick anomaly detection he model may need to be adjusted for more complicated and extensive IoT networks that support a variety of devices. More real-world testing may be necessary to assess the model's resilience against complicated, dynamic cyberattacks. Machine learning and dynamic network management can improve model prediction. The model's prediction skills can be improved by adding machine learning methods and better managing dynamic network conditions. Integrating TS-HDC with existing IoT security frameworks allows for more complete protection of the growing ecosystem.

REFRENCES

[1] Ma, D., Rosing, T. Š., & Jiao, X. (2023). Testing and enhancing adversarial robustness of hyperdimensional computing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42(11), 4052-4064.

[2] Saleem, A., Shah, S., Iftikhar, H., Zywiołek, J., & Albalawi, O. (2024). A Comprehensive Systematic Survey of IoT Protocols: Implications for Data Quality and Performance. IEEE Access.

[3] Schizas, N., Karras, A., Karras, C., & Sioutas, S. (2022). TinyML for ultra-low power AI and large scale IoT deployments: A systematic review. Future Internet, 14(12), 363.

[4] Kim, D., Yu, C., Xie, S., Chen, Y., Kim, J. Y., Kim, B., ... & Kim, T. T. H. (2022). An overview of processing-in-memory circuits for artificial intelligence and machine learning. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 12(2), 338-353.

[5] Li, Z., & Sun, H. (2023). Artificial intelligence-based spatio-temporal vision sensors: applications and prospects. Frontiers in Materials, 10, 1269992.

[6] Wang, C., Shi, G., Qiao, F., Lin, R., Wu, S., & Hu, Z. (2023). Research progress in architecture and application of RRAM with computing-in-memory. Nanoscale Advances, 5(6), 1559-1573.

[7] Li, Z., Bao, R., Zhang, W., Wang, F., Wang, J., Fang, R., ... & Shang, D. (2024). 2T2R RRAM-Based In-Memory Hyperdimensional Computing Encoder for Spatio-Temporal Signal Processing. IEEE Transactions on Circuits and Systems II: Express Briefs.

[8] Mejri, M., Amarnath, C., & Chatterjee, A. (2024). A Novel Hyperdimensional Computing Framework for Online Time Series Forecasting on the Edge. arXiv preprint arXiv:2402.01999.

[9] Chang, C. Y., Chuang, Y. C., Huang, C. T., & Wu, A. Y. (2023). Recent progress and development of hyperdimensional computing (hdc) for edge intelligence. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 13(1), 119-136.

[10] Heddes, M., Nunes, I., Givargis, T., Nicolau, A., & Veidenbaum, A. (2024). Hyperdimensional computing: a framework for stochastic computation and symbolic AI. Journal of Big Data, 11(1), 145.

[11] Yun, S., Chen, H., Masukawa, R., Barkam, H. E., Ding, A., Huang, W., ... & Imani, M. (2024). HyperSense: Accelerating Hyper-Dimensional Computing for Intelligent Sensor Data Processing. arXiv preprint arXiv:2401.10267.

[12] Zhang, T., Morris, J., Stewart, K., Lui, H. W., Khaleghi, B., Thomas, A., ... & Rosing, T. (2023). HyperSpikeASIC: Accelerating Event-Based Workloads With HyperDimensional Computing and Spiking Neural Networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42(11), 3997-4010.

[13] Katoozian, D., Hosseini-Nejad, H., & Dehaqani, M. R. A. (2024). A new approach for neural decoding by inspiring hyperdimensional computing for implantable intra-cortical BMIs. Scientific Reports, 14(1), 23291.

[14] Wang, J., Chen, L., & Al Faruque, M. A. (2023, October). DOMINO: Domain-invariant Hyperdimensional classification for multi-sensor time series data. In 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD) (pp. 1-9). IEEE.

[15] Cheng, B., Liu, J., Davis, S., Enciso, Z. M., Zhang, Y., & Cao, N. (2024, June). VAE-HDC: Efficient and Secure Hyper-dimensional Encoder Leveraging Variation Analog Entropy. In Proceedings of the 61st ACM/IEEE Design Automation Conference (pp. 1-6).

[16] Ara, I., & Kelley, B. (2024, October). Secure and Ultra-Reliable 6G IoT Algorithms with AI-Enhanced Shared Key Physical Layer Security. In 2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0148-0157). IEEE.

[17] Yu, X., Thomas, A., Moreno, I. G., Gutierrez, L., & Rosing, T. Š. (2024, May). Intelligence Beyond the Edge using Hyperdimensional Computing. In 2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) (pp. 1-13). IEEE.

[18] Hassan, E., Zou, Z., Chen, H., Imani, M., Zweiri, Y., Saleh, H., & Mohammad, B. (2024). Efficient event-based robotic grasping perception using hyperdimensional computing. Internet of Things, 26, 101207.

[19] Issa, M. A., Chen, H., Wang, J., & Imani, M. (2024). CyberRL: Brain-Inspired Reinforcement Learning for Efficient Network Intrusion Detection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[20] T. A. Mohanaprakash and D. V. Nirmalrani, “Exploration of various viewpoints in cloud computing security threats,” Journal of Theoretical and Applied Information Technology, vol. 99, no. 5, pp. 1172–1183, 2021

[21] N. Dudiki, S. Sangeetha, A. Manna, R. Pokhariyal, T. A. Mohanaprakash and A. P. Srivastava, "A Hybrid Cryptography Algorithm to Improve Cloud Computing Security," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022

[22] S. Raja, M. Parameswari, M. Vivekanandan, V. G. Krishnan, T. A. Mohanaprakash and G. Thiyagarajan, "A Systematic Analysis, Review of Data Encryption Technology and Security Measures in IoT, Big Data and Cloud," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022

[23] T. A. Mohanaprakash and J. Andrews, "Novel privacy preserving system for Cloud Data security using Signature Hashing Algorithm," 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 2019