Intelligent Multimedia Processing & Communication Systems Journal

IMPCSJ

J IMPCS (2025) 19: 21-48
DOI 10.71856/IMPCS.2025.1202812

Research Paper

A Semantic Ontology-Based Model by Ensemble Learning For Secure Attribute-Based Encryption in Fog-Enabled Smart Homes

Ronita Rezapour[1], Parvaneh Asghari [2*], Hamid Haj Seyyed Javadi [3], Shamsollah Ghanbari[4]

1. PhD Student, Department of Computer Engineering, Q.C., Islamic Azad University, Qom, Iran.
2. Assistant Professor, Department of Computer Engineering,CT.C, Islamic Azad University, Tehran, Iran. **Corresponding author*
p_asghari@iauctb.ac.ir
3. Professor, Department of Computer Engineering, Shahed University, Tehran, Iran
4. Assistant Professor, Department of Computer Engineering, ASH.C., Islamic Azad University, Ashtian, Iran

Article Info

Article history:
Received: 15 Jan 2025
Accepted: 24 Feb 2025

Keywords:
Attribute-based encryption,
Ensemble learning,
Fog-based smart buildings,
GMDH.

ABSTRACT

Fog computing empowers resource-limited applications by bringing cloud computing close to the network periphery, effectively limiting latency, improving efficiency, and assuring better resource management. Security challenges, however, restrict widespread adoption in practice, necessitating cryptographic mechanisms with low computational overheads. One well-known approach for data sharing in a secure way is Ciphertext-Policy Attribute-Based Encryption (CP-ABE), which provides a way of access control based on attributes. However, the high execution time and storage requirements of CP-ABE, due to the diversity of attributes in secret keys and access structures, limit its practicality in resource-constrained environments. In response to these problems, this paper presents a hybrid semantic model consisting of an outsourced CP-ABE with attribute revocation and an optimized AES algorithm based on ensemble learning. The model employs classifiers such as GMDH, KNN, and SVM to identify attributes relevant to CP-ABE. Additionally, the Dragonfly optimization algorithm and ontology-based semantic techniques enhance the efficiency of feature selection. With experimental analysis on five smart building datasets, the prediction performance of this model outperforms existing methods. The times of encryption, decryption, and attribute revocation decreased to 2.99 ms, 2.86 ms, and 18.6 ms. The growth of storage for secret keys and access structures was reduced to 13.56 KB and 10.4 KB, which made its use more efficient and secure. Overall, the results indicate that the model improves data security and minimizes computational overhead, leading to a more feasible implementation of CP-ABE for fog computing scenarios.

NOMENCLATURE

t_i	Tripping time of the *i*-th relay
TDS_i	Time dial setting (TDS) of the i-th relay
$I_{F,i}$	Short circuit current of the i-th relay
$I_{p,i}$	Pickup current setting (PCS) of the i-th relay
$I_{H,i}$	Breakpoint of the i-th relay's dual-setting curve
A_i, B_i	Standard relay curve coefficients of the i-th relay
TDS_i^f	TDS of the first level of the i-th relay
TDS_i^s	TDS of the second level of the i-th relay
$I_{p,i}^f$	PCS of the first level of the i-th dual-setting curve's relay
$I_{p,i}^s$	PCS of the second level of the i-th dual-setting curve's relay

Publisher: Islamic Azad University – Zanjan Branch

I. Introduction

With the growing penetration of information technology and automation into critical infrastructures, the intelligent environments and smart buildings have sparked much concern. This transformation has been greatly supported by computational advancements [1]. With the advancements of modern data-driven technologies, smart environments not only provide deeper integration but also more optimized data sharing and more accurate decision making, resulting in higher efficiency, reduced costs and better operational effectiveness [1], [2].

However, most past models of smart environments rely on cloud computing as a backbone. Nevertheless, smart buildings and intelligent systems generate enormous amounts of data that result in the the explosive growth of data, while traditional cloud cannot provide fast responses to heterogeneous, latency-sensitive services [6]. To overcome these constraints, fog computing has been introduced aiming at distributed management of data along with its effective processing in real-time [1[, [6], [7]. In fog-aided models, data received from smart devices are processed locally, and are sent to the cloud for storage. This approach eliminates latency, enhances mobility and location awareness, and minimizes data transmission costs [6], [7], [8].

While this is convenient, fog computing comes with issues like limited computational resources and data security. Data offloading is commonly needed to upload to fog nodes and cloud storage due to limited processing and storage capabilities of smart devices. As a result, data owners lose direct control over their data, which introduces security risks and makes access control difficult. Therefore, security has turned into one of the major concerns in fog-driven smart buildings [6], [7], [8], [9]. To address these issues, cryptographic techniques provide an effective solution allowing data to be securely stored in untrusted servers that can reduce security risks and uphold user privacy [4], [5], [6], [7], [8], [9], [10].

In this work, we use the attribute-based encryption (ABE) for fine-grained access control in fog-assisted smart scenarios. ABE allows data owners to create an access policy over descriptive attributes tied to their encrypted data and to the secret keys of users [2], [4], [5], [9]. Such attributes, describing data properties, user roles, and domain-sensitive properties, are crucial to enacting security policies [2], [4], [11]. Ciphertext-policy attribute-based encryption (CP-ABE), the first of the two central sorts of ABE, enables the information owner to set access policies to guarantee that only legitimate users can decrypt the data. Despite CP-ABE providing security augmentations, its implementation introduces several challenges with regards to encryption and decryption execution time, attribute revocation, and storage overhead [9], [10], [11], [12], [13].

However, the cryptographic operations demand more and more resources with the increase of the attributes and complexity of the access structures. Further, as the count of cryptographic inferences goes up, the execution time for encryption and decryption is directly proportional to it, thereby leading to a decrease in system efficiency [4], [9], [11]. Moreover, attribute revocation is another big problem, since any access structure modifying for the revoked attributes may lead to many users requiring frequent re-encryption and key updates [4], [9], [10], [11], [12], [13]. This shapes up computation and storage costs, making policy enforcement more challenging in fog environments.

In this study, we introduce a new semantic-based approach thanks to ontology modeling that can be used to convert simple attributes into functional and semantic attributes required in cryptography as a solution for these issues. This approach, through suppressing the diversity of attributes, can make access policies more interpretable, reduce the computation complexity and the time required for cryptographic operations, and improve the efficiency of cryptography [5]. Furthermore, the suggested scheme optimizes storage and policy management since it achieves a smaller ciphertext and simpler policy structure and minimizes the number of times access policies need to be updated [14], [15], [16], [17].

In this research, we introduce an enhanced security model for fog-based smart buildings by integrating outsourced CP-ABE (Ciphertext-Policy Attribute-Based Encryption) with attribute revocation and proxy re-encryption (PRE) to minimize communication costs and strengthen security. The model utilizes the Advanced Encryption Standard (AES) to optimize encryption and decryption efficiency. Additionally, by incorporating a semantic ontology-based approach in conjunction with the Dragonfly optimization algorithm through ensemble learning, the framework effectively selects and prioritizes semantic features, which reduces storage requirements and enhances system performance. By addressing critical challenges in fog-based smart environments—such as security threats, cryptographic execution time, complexity of attribute revocation, and storage overhead—this study aims to improve system efficiency, reduce latency, and enhance the quality of service in intelligent infrastructures.

Efficient, scalable attribute and access control policy management with heterogeneous IoT data and dynamic environmental changes is another major challenge in CP-ABE for smart homes. Therefore, this paper addresses these challenges by offering a semantic representation of user attributes, devices, and data using a semantic ontology-based model with ensemble learning and attribute-based encryption. This helps enhance policy processing, mitigates computational cost, and allows dynamic, context-sensitive adjustment of access control. Fig 1 illustrates the security framework designed for a fog-based smart building. This study makes several contributions, which can be summarized as follows:

1. To the best of our knowledge, this study is the first to optimize the number of attributes used in different phases of a CP-ABE scheme by leveraging ensemble learning to select the most relevant semantic features further enhances the cryptographic efficiency.
2. This research integrates the Dragonfly optimization algorithm with a KNN-based core and an ontology-driven approach to identify the most effective semantic features to enhance the cryptographic efficiency.
3. Our approach employs ensemble learning technique, including GMDH, SVM, and KNN, to find out the most optimal attributes for CP-ABE to have secured communication in fog environments. This approach not only strengthens security but also enhances encryption and decryption efficiency while reducing the error rate.
4. The introduced model combines CP-ABE with AES symmetric encryption for better security and computation accuracy. The scheme is used to optimize resource utilization in distributed environments by outsourcing complex encryption and decryption operations to fog nodes.
5. Our proposed scheme uses a reduced number of semantic attributes in order to reduce the costs of encryption, decryption, key generation, ciphertext size and access policy processing. This optimization results in improved runtimes and storage efficiency across a wide range of domains, from fog to cloud systems, while maintaining strong security assurances. We evaluate our model to assess its performance and effectiveness.

The rest of this paper is organized as follows: Section II provides a review of related work. Section III presents the foundational background necessary for this research. Section IV outlines the system architecture and security framework. Section V describes the proposed approach, detailing its components and structure. Section VI focuses on the security analysis and performance evaluation of the proposed model. Finally, Section 7 concludes the study.

II. Related Work

This section discusses the relevant literature related to our work, specifically on aspect-based explanation design, ontology, feature selection, and ensemble classifier.

In [13] a multi-authority CP-ABE-based data-sharing and encryption scheme along with a robust user revocation mechanism were proposed in order to fulfill the security requirements of smart grids in fog environment. The study in [7] introduced a multi-authority CP-ABE scheme that focused on the reduction of both private key and ciphertext storage. In [11], a model outsourced the decryption process using fog through the Bloom filter technique. The paper proposes a multi-authority CP-ABE for fog computing to allow group keys to support attribute revocation and access management in [15]. The work in [4] presented a decentralized CP-ABE framework that identified and eliminated malicious users via a white box tracking mechanism and further removed them using the KUNodes method. The authors in [9] proposed a CP-ABE design for fog-assisted E-health systems, in which they exploit the PRE technology to realize the efficient control and revocation over medical data access attributes. Likewise, [18] introduced the ontology-based security architecture for cloud-based electronic health record systems and highlighted the need for scalability and semantic access control in multi-authority ABE encryption.

In [19], A model was created to identify lung cancer with KNN and genetic algorithms, optimizing the feature selection to minimize the dataset dimensionality. An anomaly detection-based intrusion detection mechanism applicable to IoT-enabled smart environments is studied in [20]. The work in [21] proposed a security model for dirty power grid attacks. A lightweight and dynamic ontology model was proposed in [22] to improve semantic interactions among IoT services, along with a machine learning technique for concept extension. The study in [23] adopted an ontology-based approach to energy management and cost reduction in smart homes. The research in [24] presented an ensemble learning-based framework for smart grid energy consumption prediction using XGBoost, SVR, and KNN regression. In this context, [25] proposed an automated dietary assessment system based upon ADLs based on ensemble learning for weight management, whereas [26] created a method used for recognizing daily activities in smart homes using data from a wearable device.

Tables I and II summarize the reviewed studies on CP-ABE, ontology applications, feature selection, and classification. The analysis indicates that there has been no previous attempt integrating a semantic ensemble model utilizing ontology and machine learning to build the security and efficiency of fog-based systems. Nor has a semantic-ensemble-based approach been applied in previous studies to establish attributes for ABE. Existing techniques, when run independently, cannot guarantee high security with low performance metrics for the sensitive environments of fog computing.

III. Background

This section presents the foundational concepts utilized in our research. The key definitions and notations are provided in Table III.

A. Bilinear maps

The sets G_1, G_2, and G_T are cyclic groups of prime order p, where g_1 and g_2 are generators of G_1 and G_2, respectively. A bilinear map $\hat{e}\colon G_1 x G_2 \rightarrow G_T$ is recognized as valid when the following conditions are satisfied. In this case, the quintuple $(g_1, g_2, G, G_T, \hat{e})$ is referred to as a bilinear group:

-Bilinearity: $\hat{e}({g_1}^{a}, {g_2}^{b}) = \hat{e}(g_1, g_2)^{ab},\ \forall\, g_1, g_2 \in G\ \ \&\, a \in \mathbb{Z}_p, b \in \mathbb{Z}_p$.

-Non degeneracy: $\hat{e}(g_1, g_2) \neq 1$.

-Computability: The value of $\hat{e}(g_1, g_2)$ can be computed in probabilistic polynomial time.

B. Access structure

In the proposed model, the access structure A is defined over a set of elements $\{p_1, p_2, \dots, p_n\}$, where A consists of a collection of non-empty subsets of these elements such that $A \subset 2^{\{p_1, p_2, \dots, p_n\}} \backslash \{\emptyset\}$. If A is monotonic, then for all sets $B, C \in A$, if $B \subset C$, it follows that $C \in A$. In this study, we examine a monotonic access structure, which includes the set of permissible feature sets.

C. Linear secret sharing schema

In CP-ABE schemes with high expressiveness, the matrix of LSSS is commonly used for representing monotone access structures. A secret sharing scheme $\prod$ over a group of participants ρ is considered linear if the following conditions hold:

- The participants‘ shares can be seen as forming a vector in $\mathbb{Z}_p$.
- To $\prod$, a matrix M of size $l * n$ exists, with ρ(i) mapping the i-th row of the matrix to the respective participant, and $i = 1, 2, \dots, l$. The vector $M\vec{v}$ has l shares corresponding to the secret s, defined by $\prod$. Here, $v = (s, r_2, r_3, \dots, r_n)$ is of order n, with $r_2, r_3, \dots$, $r_m \in \mathbb{Z}_p$ randomly selected, while $s \in \mathbb{Z}_p$ represents the secret to be shared. The share corresponding to the party $p(i)$ is $(M\vec{v})_i$ [24].

If the access policy A is in LSSS representing an authorized set S, and if $I \subset \{1, 2, \dots, l\}$ is identified as $I = \{i\colon p(i) \in\ S\}$, and the value of $\lambda_i$is representing valid shares of the secret s according to $\prod$, there must be coefficients $\{\{w_i \in \mathbb{Z}_p\}, i\ \in I$ such that $\left\{\sum_{i \in I_s} w_i\, \lambda_i = s\right\}$ is polynomial time compmlexity computable. Thus, it is obvious that $\lambda_i = (M\vec{v})_i$ [24].

D. Decisional Q-Parallel BDHE assumption

Considering the bilinear map parameters on an elliptic curve, we define the set $(p, g, G, G_T, \hat{e})$. If an adversary A obtains the following set: $\vec{y} = g, g^s,\ g^{\mathrm{a}}, g^{a^2}, \dots, g^{a^{\mathrm{q}}}, g^{a^{\mathrm{q}+2}}, \dots, g^{a^{2\mathrm{q}}}$, $g^{\mathrm{s.b_j}},\ g^{\mathrm{a/b_j}},\ g^{a^2/\mathrm{b_j}}, \dots, g^{a^{\mathrm{q}}/\mathrm{b_j}}, g^{a^{\mathrm{q}}+2/\mathrm{b_j}}, \dots, g^{a^{2\mathrm{q}}/\mathrm{b_j}}$,

$1 \leq j \leq q, g^{\mathrm{a.s.b_i/b_j}},\ g^{a^2\mathrm{.s.b_i/b_j}}, .., g^{a^{\mathrm{q}}\mathrm{.s.}\frac{\mathrm{b_i}}{\mathrm{b_j}}}, 1 \leq i\ \neq j \leq q$

If the values $q\ +\ 2,\ \in Z_p$, and the random elements $a, s, b_1, b_2, \dots, b_q, \in Z_p$ are considered, then, if the adversary cannot probabilistically decide in polynomial time whether $\hat{e}(g, g)^{sa^{q+1}}$ is distinguishable from a random element in G_T, the q-parallel BDHE assumption holds. Specifically, the advantage $Adv_A^q(\lambda)$ of adversary in solving the decisional q-parallel BDHE problem with respect to λ as a security parameter is represented as follows [9, 25, 2].

$$Adv_A^q(\lambda) = \Big|Pr\big[A(\ \vec{y}, \hat{e}(g, g)^{sa^{q+1}}\ \ ) = 1\,\big] \\ - Pr\left[A\left(\vec{y}, R \overset{R}{\leftarrow} G_T\right) \\ = 0\right]\Big|$$

IV. System and security model

In this section, we first present the architecture of the proposed system, followed by an analysis of its security model.

A. System model

Fig. 2 depicts the system architecture of the proposed model, detailing the following components: cloud service provider, ontology-based semantic feature selection, attribute authority, fog node, IoT device, and data user.

1) Cloud Service Provider

The Cloud Service Provider (CSP) serves as a powerful computational center where users can upload and share all their stored data publicly. It provides direct storage of data transmitted over fog nodes in encrypted ciphertext from IoT devices. CSP sends the encrypted data to the appropriate fog node upon the request of an authorized user. In case some attributes need to be revoked, the CSP additionally re-encrypts the affected ciphertexts to ensure that the outdated/unauthorized access is indeed revoked.

2) Attribute Authority

The Attribute Authority (*AA*) is a fully trusted third party that initializes the system, set up public parameters, and creates necessary cryptographic keys such as master secret keys (*MSK*), secret keys (*SK*), and symmetric session keys (*K*). Beyond these foundational tasks, the *AA* applies an ontology-based semantic feature selection mechanism to enhance the system.

3) Semantic ontology-based feature selection

The semantic method of the attribute authority embeds the Dragonfly feature selection algorithm, enhanced by an ontology and supported by ensemble learning techniques. Using this generalised method, the objective of generating unique semantic and semantic attributes that are paramount in determining the policy structures and generating secret keys is aimed at. The use of the Dragonfly algorithm enables efficient selection of features, and the ontology offers a logical structure that can be used to determine the relationships between attributes. By combining the advantages of multiple algorithms, ensemble techniques provide even more strength by improving the robustness and reducing the error in the identification of attributes in the resulting policies as well as in the key generation process.

4) Fog node

Semi-trusted Fog Nodes (FNs) play an integral role in this fog-based architecture as intermediates which communicate the Cloud Service Provider (CSP) and end-users. These nodes are usually small routers or servers, deployed near the edge of the network and close to the IoT devices. Fog nodes can send and receive encrypted data, and most of the encryption and decryption work is performed by these nodes.

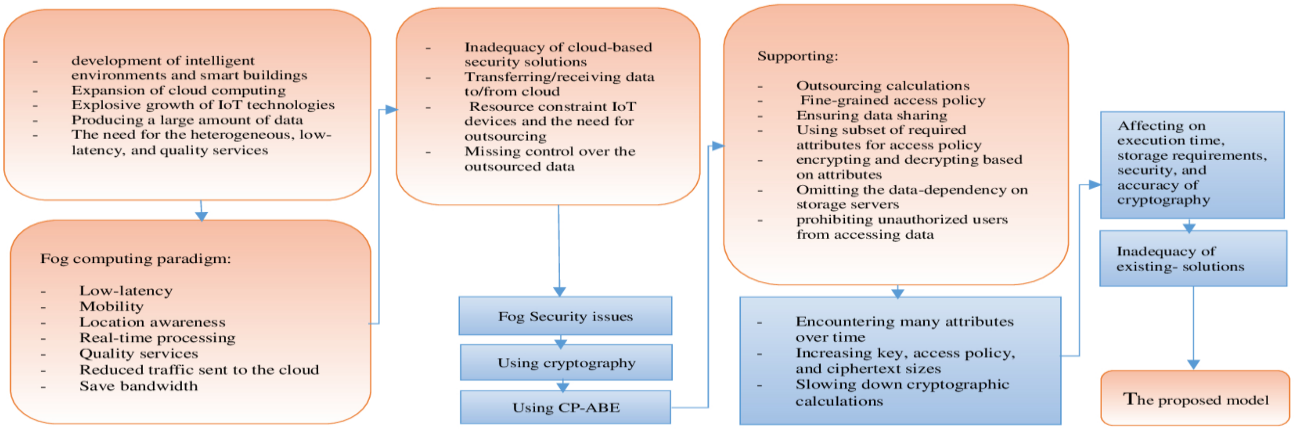
- development of intelligent environments and smart buildings
- Expansion of cloud computing
- Explosive growth of IoT technologies
- Producing a large amount of data
- The need for the heterogeneous, low-latency, and quality services
Fog computing paradigm:
- Low-latency
- Mobility
- Location awareness
- Real-time processing
- Quality services
- Reduced traffic sent to the cloud
- Save bandwidth
- Inadequacy of cloud-based security solutions
- Transferring/receiving data to/from cloud
- Resource constraint IoT devices and the need for outsourcing
- Missing control over the outsourced data
Fog Security issues
Using cryptography
Using CP-ABE
Supporting:
- Outsourcing calculations
- Fine-grained access policy
- Ensuring data sharing
- Using subset of required attributes for access policy
- encrypting and decrypting based on attributes
- Omitting the data-dependency on storage servers
- prohibiting unauthorized users from accessing data
- Encountering many attributes over time
- Increasing key, access policy, and ciphertext sizes
- Slowing down cryptographic calculations
Affecting on execution time, storage requirements, security, and accuracy of cryptography
Inadequacy of existing- solutions
The proposed model

Fig. 1. Security scenario of a smart building based on fog computing.

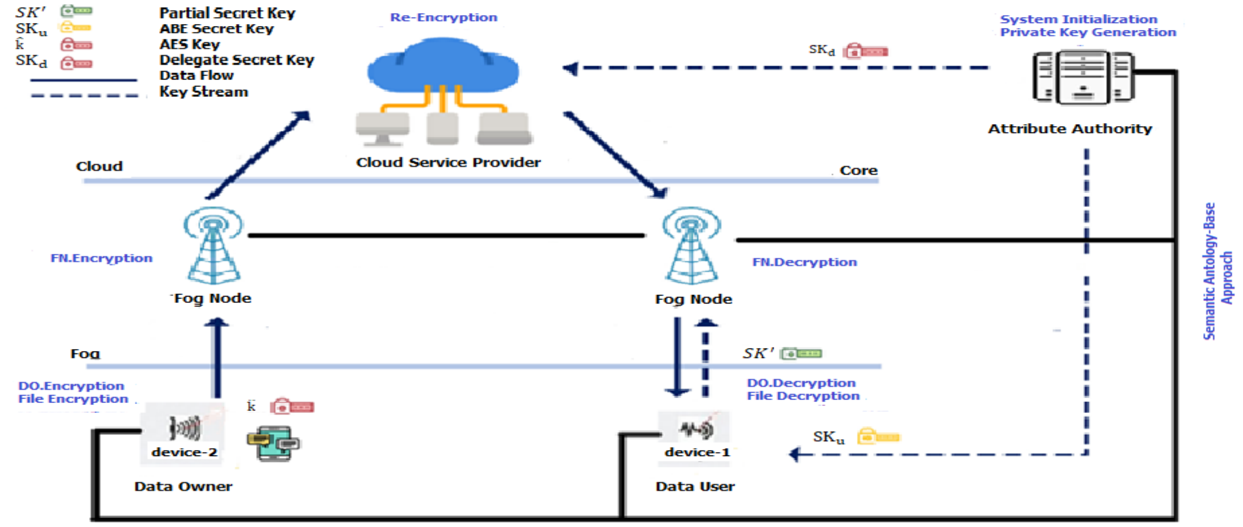

Fig. 2. The architecture of the proposed model.

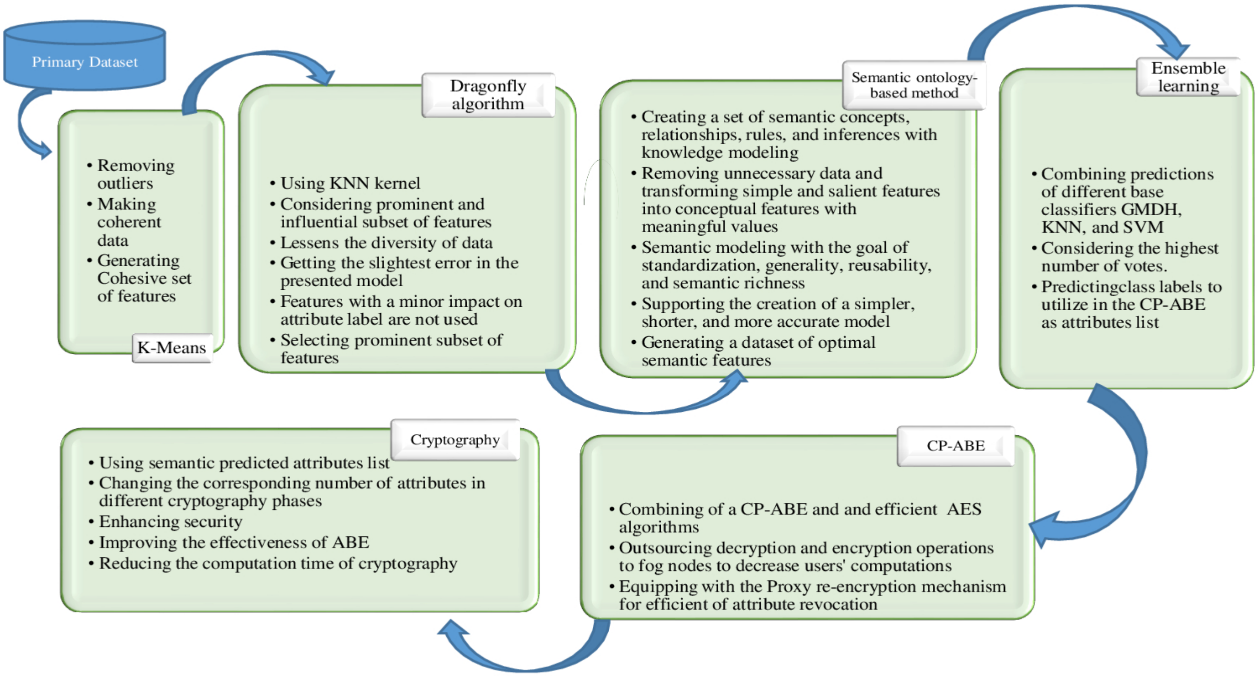
Primary Dataset
• Removing outliers
• Making coherent data
• Generating Cohesive set of features
K-Means
Dragonfly algorithm
• Using KNN kernel
• Considering prominent and influential subset of features
• Lessens the diversity of data
• Getting the slightest error in the presented model
• Features with a minor impact on attribute label are not used
• Selecting prominent subset of features
Semantic ontology-based method
• Creating a set of semantic concepts, relationships, rules, and inferences with knowledge modeling
• Removing unnecessary data and transforming simple and salient features into conceptual features with meaningful values
• Semantic modeling with the goal of standardization, generality, reusability, and semantic richness
• Supporting the creation of a simpler, shorter, and more accurate model
• Generating a dataset of optimal semantic features
Ensemble learning
• Combining predictions of different base classifiers GMDH, KNN, and SVM
• Considering the highest number of votes.
• Predictingclass labels to utilize in the CP-ABE as attributes list
Cryptography
• Using semantic predicted attributes list
• Changing the corresponding number of attributes in different cryptography phases
• Enhancing security
• Improving the effectiveness of ABE
• Reducing the computation time of cryptography
CP-ABE
• Combining of a CP-ABE and and efficient AES algorithms
• Outsourcing decryption and encryption operations to fog nodes to decrease users' computations
• Equipping with the Proxy re-encryption mechanism for efficient of attribute revocation

Fig. 3. Our approach overview.

TABLE I. Relevant studies on CP-ABE and ontology

Ref	Year	Key Purpose	Technique	Strengths	Weaknesses
[13]	2022	Ensuring verifiable outsourcing of decrypting and data sharing in CP-ABE for secure revocation in fog-based smart grids	Multi-authority framework for decentralized access control; Outsourcing cryptographic to fog nodes; User revocation through versioning of keys and ciphertexts; Minimizing attributes by optimizing key and ciphertext structures	Optimized computational overhead in cryptographic operations; Delegated decryption and data-sharing system with verifiability; Compatibility with LSSS policy	Reduced efficiency; Increased key updating costs; High costs associated for multi-authorization
[7]	2022	Enhancing the performance of ABE for resource-limited devices in fog environments	Multi-authority framework; Non-dependence on the size of secret keys and ciphertexts	Effective cryptographic processes; Optimized length of keys and ciphertext; Efficient data-sharing for IoT	Incompatibility with LSSS policy; Complexity in multi-authority calculations
[11]	2022	Fog computing CP-ABE decryption efficiency improvements	Collaborative decryption in groups; Bloom filter and cooperation functions for attribute management	Minimizing storage requirements and power consumption in IoT; Optimized computational overhead for IoT and fog; Privacy-preserving access control policies	Not supporting LSSS policy; Difficulty in managing keys; Necessity of MA architecture; Lack of load balancing for attribute allocation to fog nodes; No support outsourcing encryption
[18]	2022	Rich semantic access control in cloud-based healthcare encryption systems with multiple authorities	Multi-authority mechanism; Ontology-based semantic access control instead of traditional access control	Enhanced security through a semantic approach in access policies; Scalable access management for a large number of users and organizations	Elevated computational costs due to multi-referencing; Scalability challenges with increasing users and organizations at a global level
[15]	2021	MA-CP-ABE enabling dynamic user attribute modifications and delegating computations to fog nodes	Multi-authority framework; Fog delegating cryptographic tasks; Group key components for flexible adjustments of attributes; Revoking attribute using binary tree structures	Minimal processing complexity; Enhanced performance and robustness for fog-enabled applications; Reduced runtime for cryptographic computations; Compatibility with LSSS policy	Not accounting for communication latency and energy usage; No investigation into storage requirements for cryptographic computations
[4]	2021	Effective malicious user detection and user revocation in a multi-authority CP-ABE cloud system	White box tracking for detecting malicious users; Delegating decryption tasks	Compatibility with LSSS and diverse cryptographic capabilities; Reasonable computational cost	High computational overhead and execution time for cryptography; Lack of encryption outsourcing; No implementation of black-box traceability
[9]	2021	Enhancing access policy modification and performance in fog-enabled CP-ABE	Delegating decryption/encryption; Proxy-based re-encryption for effective attribute revocation	Less computation time and storage needs; Shortened time to revoke attribute; Compatibility with LSSS policies	Incompatible with multi-authority systems; Lacks tracking capabilities

TABLE II. Relevant studies on feature selection, classification techniques and other applications of ontology.

Ref	Year	Key Purpose	Technique	Strengths	Weaknesses
[20]	2023	A framework for Intrusion detection in IoT-enabled smart environments using anomaly detection strategies	AdaBoost algorithm for ensemble learning; Integration of multiple feature selection methods such as Boruta in Random Forest, Mutual Information, and Pearson Correlation Coefficient; K-fold cross-validation for assessment	Improved accuracy of Intrusion Detection Systems; Reduced learning/detection time; Enhanced performance and attack resistance	Lack of consideration for data diversity and multi-class classification; Requirement for lightweight learning models in resource-constrained IoT devices
[21]	2022	An intelligent IDS for precise attack classification in intelligent power grids utilizing binary gray Wolf optimization-based feature selection	Ensemble classification; Adjusting learning mechanisms using multiple meta-parameters; Self-tuning Bayesian optimization method	Minimal prediction processing time; Effectiveness for real-time Intrusion Detection applications; Superior accuracy and reliability in 2-class and 3-class classification tasks; Decreased misclassification rate	Longer training phase in ensemble models incorporating multiple classifiers; Additional time required to set meta-parameters; Inability to handle multiple-category classification challenges
[19]	2021	A hybrid approach for optimal feature selection and data reduction to enhance cancer diagnosis and classification efficiency	KNN method; Genetic algorithm (GA)	Superior model accuracy; Optimized execution time of the scheme	Performance reliance on data quality; Limited exploration of impactful feature selection techniques
[22]	2021	A lightweight dynamic ontology model for IoT	Machine learning for ontology expansion; Dynamic lightweight ontology	High flexibility and scalability; Reduced resource consumption and response time; Accuracy in semantic interaction	Implementation complexity for compatibility with various devices; Need for high-quality input during training
[23]	2021	An ontology-driven strategy for energy optimization in smart homes	Utilization of Data Analysis Techniques; Integration of Energy Management Systems and Ontology	Automated Analysis and Decision-Making; Optimizing Energy Consumption and Smart Management; High flexibility and scalability	Need for accurate data for optimal system performance; Complexity in ontology design
[24]	2020	A precise energy consumption prediction model in smart grids leveraging ensemble learning for effective resource management	Hybrid ensemble learning approach incorporating XGBoost, Support Vector Regression, and K-Nearest Neighbors; Genetic algorithm-based optimization	Reduced error rate; Enhanced precision of the ensemble model	Reduced performance with high-dimensional data; Inability to fulfill all contextual variables; Inaccurate results in certain metrics
[25]	2020	A diet evaluation framework for weight loss management incorporating image recognition and ensemble learning with a voting approach	Ensemble learning setup using VGGNet, GoogleNet, ResNet, and InceptionV3; Integrating techniques: max, min, ave, med, and weighted softmax probabilities in the learning process; Weighted voting via Bayesian optimization; Learning configurations with a convolutional neural network pre-trained	Reduced model training time; Enhanced adaptability and efficiency in the vote-driven classification approach; Improved accuracy	Limited support for personalizing user data images and handling diverse data; Increased hardware costs for CPU/GPU-level tasks and memory computing; Necessity for fine-tuning operations for optimal performance

. TABLE III. Basic definitions and notational conventions.

Variable	Explanation	Variable	Explanation
DA		SVM	
m	Accuracy of classification	w	Weight array
n	Feature dimensionality reduction	b	Bias vector
$\|R\|$	Subset feature count	ξ_i	Relaxing factor
$\|C\|$	Dataset feature count	K_{ebf}	Kernel
Ontology		c	Cost of incorrect classification
$(swrlb: lessThan(?\,v, x))$	With a value less than x for v	ɣ	Gamma : Kernel non-linearity
$(swrlb: greaterThan(?\,v, x))$	With a value greater than x for v	x_i	Data sample
SystemCommand(?cmd)	Sending the command cmd	y_i	Sample category
$KeyFeature(?\,p)$	Considering the feature p as a key feature	CP-ABE	
		λ	The security factor in attribute-based encryption
$correlation(?\,p_1, ?\ p_2, ?\,cor)$	Correlation between two features Correlation between two features (?p1, ?p2)	S	The potential attributes in the semantic attribute set
		$[l]$	The collection {1, 2, …L}
$RemoveFeature(?\,p_2)$	Removal of feature p_2	$a \overset{R}{\leftarrow} \mathbb{Z}_p$	Randomly selecting elements a with uniform distribution from $\mathbb{Z}_p$
$valueVariance(?\,p, ?\,v)$	Value changes of feature p based on v	(A, p)	The access structure represented by the matrix A
$isHighlyVariable(?\,p, true)$	Significant changes in feature p	$p(i)$	The row i of policy matrix
GHMD		$\hat{k}$	Symmetric key: , $\hat{k} \overset{R}{\leftarrow} G_T$
RMSE	Root mean square error	$\vec{v}$	A random horizontal vector $\vec{v} \in \mathbb{Z}_p^n$
$RMSE_{min}$	Optimal RMSE	y	The attribute y scheduled for revocation
$RMSE_{max}$	Worst RMSE	$h(x)$	H $\{0,1\}^* \rightarrow$ G represents a hash function
MSE	Mean absolute error	$\{SOD\}_k$	Collected data encrypted symmetrically under key $\hat{k}$
e_c	Selection-pressure	CT_0	The initial ciphertext generated from ABE by fog
a	Custom value within the range [0, 1]	TCT	Temporary encypted data of $\hat{k}$
y_i	Estimated parameter value (i)	CT_1	The ciphertext re-encrypted version
G_i	Observed parameter value (i)	PP	Public parameter
n	observation count	MSK	Mask secter key
KNN		SK	Secret key
K	KNN neighbor count	SK_d	Delegation key
x_i	Training sample(i)	$\hat{F}$	The intermediate ciphertext during decryption by fog
x_i'	Test sample(i)	$\mathbb{Z}_p$	A set of integers in modulo p residue class
n	Features count	G	A bilinear algebraic group defined by prime order, generated by g

5) Data owner

According to proposed framework sensing and communication functional integrated IoT devices are referred as Data Owners (DO). These DOs are responsible for picking the attributes that define access control policies and encrypt data with defined policy.

6) Data user

Data Users (*DUs*) are those IoT devices or people accessing the data found in the cloud data center. while the data user requests data access and if his attributes satisfies the access policy, he completes a segment of the decryption process and gets the required information.

B. Selective security model

The security model of the proposed system is framed as a security game between an adversary and a challenger as described in [9]. In this game, the adversary must guess which attributes in the challenge structure to target. This is described as follows [2], [9], [26]:

Initialization: The adversary, denoted as A, sends a challenging access structure to the challenger, C. ***Setup:*** The challenger, C, executes the Setup algorithm and sends the public parameters to adversary A. ***Phase 1:*** Adversary A selects a set of attributes and sends requests for private keys corresponding to these attributes, as long as they do not participate in the challenge access structure.

Challenge: Adversary A sends two messages, M_0 and M_1, of equivalent length to challenger C. The challenger then randomly selects a value $\mathrm{b} \in \{0,1\}$, encrypts message M_b according to the access structure, and returns the corresponding ciphertext back to A. ***Phase 2:*** Adversary A repeats the cycle from *Phase 1*. ***Guess :*** To win, the adversary must guess: if the adversary guesses for the value which is $b' = b$. We define The adversary's advantage in attacking the scheme as $|Pr[b' = b\,] - ½|$.

V. Proposed Approach

In this section, we begin by presenting a general overview of our work, followed by an in-depth discussion of the different elements that make up our approach.

A. Approach overview

The objective of the proposed model is to reduce the number of relevant attributes during the multiple steps of CP-ABE by coupling with a semantic model based on ontology. This study aims to enhance cryptographic performance by accelerating execution time, minimizing storage need, and improving the system's security measures. It also reduces latency and improves the quality of service in fog-based smart building environments. There are two core components of our model:

semantic ontology-based feature selection and attribute-based encryption. A visual summary of our approach is shown in Fig. 3. The class labels that are predicted by semantic ensemble learning are considered to be lists of attributes that are fed into the cryptography process enabling efficiency and security in ABE (Attribute-Based Encryption). This is advantageous because the number of attributes that need to be considered affects both computational time and storage requirements.

The CP-ABE key generation and update phase usually require processing time and storage that grows with the size of the key. Likewise, the running time and storage requirements in the encryption phase are also linear with the number of attributes that are specified in the access policy. In the local and outsourced decryption phases, the decryption time and storage used depend linearly on the number of attributes required for decryption. Moreover, for the ciphertext update phase, the time and storage requirements are also affected by the number of access control attributes. By controlling relevant features across the cryptographic pipeline, we hope to achieve performance along with improved security.

We evaluated our cryptographic model on various datasets such as, HomeC, OSH, SBS, IEQ, and CU-BEMS against the scheme presented in [9] by comparing the execution time and storage requirements. For instance, in a smart environment, a device (data owner) collects data (temperature, humidity, etc.) and shares it with specific devices (data users), this sensitive data must be encrypted by the data owner before uploading to the cloud. Table IV summarizes a shared feature set that includes name, date of manufacture, file size, role, department, topic, temperature, light and motion sensor presence, and light and temperature sensor data.

The proposed model also generates semantic feature labels, which are subsequently utilized in CP-ABE for constructing both the access policy and the user's private key. The semantic features are aligned with the semantic model via the Dragonfly optimization algorithm, which integrates ontology-based rules and semantic relationships, further enhanced through ensemble learning techniques. The specific details regarding these semantic features and their alignment are outlined in Tables V and VI. When a device enters the system, it is assigned specific features, illustrated by the attribute collections for two devices: *Device 1= {Control Service, Disabling Trigger, Portable}* and *Device 2= {General Service, Archiving Trigger, Non-Portable}*.

As illustrated in Fig. 4, the owner of data specifies the access policy *{"Control Service" AND "Portable"}* and performs encryption in accordance with this policy. According to this access policy, the attributes of Device1 that align with its secret key fulfill the conditions, granting access to the shared data. On the other hand, Device 2 is denied access. Device 1 decrypts the ciphertext using its private key whenever it needs access to the encrypted data (ciphertext). In the event of attribute revocation, the update of private keys is performed, while the cloud service provider is responsible for re-encrypting the data the relevant ciphertext. Upon revocation of an attribute, a section of the ciphertext undergoes decryption, and the new access policy is enforced. This operation is performed based on the identity of the data owner.

B. Approach details

In this section, we provide a comprehensive explanation of our model.

1) Preprocessing using the K-Means

The raw data undergoes a preprocessing phase in which outliers are eliminated, and the data is transformed into a format suitable for further analysis. This step is vital, as real-world datasets often contain redundancies, missing values, and NaN entries. A well-prepared dataset is necessary for obtaining reliable training results. Preprocessing helps in making the data more consistent, simplifying the model, speeding up sample processing, and improving feature selection accuracy [30]. By utilizing the K-Means-based clustering technique, the model can categorize the data, eliminate outliers, and discard weak clusters [30]. The K-Means algorithm was chosen for its computational efficiency and ability to manage large, high-dimensional datasets. This method is simple yet effective for categorizing data with diverse characteristics [31]. Following this, the data is normalized to enhance accuracy and achieve superior results.

2) Dragonfly optimization algorithm

The Dragonfly Algorithm (DA) chooses optimal features from a feature set. This algorithm is a stochastic search technique that mimics the behavior of dragonflies in nature [30], [32]. Based on the criteria of this algorithm, the most effective and efficient collection of features, which contribute to accuracy of classification, is chosen. This method eliminates irrelevant features and identifies significant ones that influence classification performance [33]. In this study, we used the Dragonfly algorithm due to its notable advantages, enabling it to quickly reach the global optimum and implement it with greater precision [34]. The DA effectively balances exploration and exploitation to handle challenges in complex search spaces, similar to real-world scenarios [35]. Its advantages over other algorithms include transparency, faster search times, and ease of improvement and hybridization. Moreover, compared to other algorithms, it has a better capability to select features that provide more informative data [34], [35]. We empirically adjusted the exploitation and exploration processes and fine-tuned the group parameters (a, c, e, f, s, w) during optimization.

In the Dragonfly algorithm, the objective function is specified by equation 1. Feature selection is inherently a bi-objective problem, where one goal is to minimize the size of feature set, and the other is to reduce classification error. The features that can minimize the objective function are assessed for their suitability. Better capability to select features that provide more informative data [34[, [35]. We empirically adjusted the exploitation and exploration processes and fine-tuned the group parameters (a, c, e, f, s, w) during optimization.

In the Dragonfly algorithm, the objective function is specified by equation 1. Feature selection is inherently a bi-objective problem, where one goal is to minimize the size of feature set, and the other is to reduce classification error. The features that can minimize the objective function are assessed for their suitability.

$$DA_{Fitness} = \min\left[m \times kfoldLoss + n \times \frac{|R|}{|C|}\right] \quad (1)$$

In this context, $m + n = 1$, and kfoldLoss represents the classification error obtained from the K Nearest Neighbor (KNN) algorithm during cross-validation. In the KNN algorithm, classification is based on the distance between test and training samples [19]. During the execution of K-fold cross-validation, the original dataset is randomly divided into K subsets; one subset is used for testing, and the remaining *K-1* subsets are used for training. The K-fold cross-validation process calculates the error (using mean squared error) of the regression model CVMdl. The kfoldLoss computes the error for each validation subset based on a model trained on the corresponding training subset. The model undergoes K evaluations, and the average result is taken. In the KNN algorithm, the Euclidean distance metric is used, and the number of neighbors is set to 5. The K-Fold cross-validation process consists of 10 splits. Furthermore, $|R|$ and $|C|$ refer to the number of features in the subset and the entire dataset, in the same order. The parameters m and n govern the significance of classification accuracy and feature reduction, with values $m = 0.99$ = 0.99 and $n = 0.01$ in this study. As a result, the algorithm is capable of selecting key features. Finally, the optimized dataset is sent to the semantic model based on ontology to perform the conceptual reformation of the data. Table VII presents the values of the parameters used in the DA algorithm.

3) Ontology-based Semantic Approach

Ontology, in the field of computer science, is a formal representation of knowledge as a set of concepts, categories, and the relationships between them, that can be useful for the purpose of simplifying and enriching information. Ontologies allow raw data to be converted into well-structured and meaningful information, enabling not only better processing and retrieval of data, but also integration of heterogeneous datasets [36]. In the context of a complex ecosystem, like IoT and fog computing, ontologies aid in ensuring a shared understanding of the data through an accurate representation of the attributes and their integration among components, which promotes intelligence [37].

As smart buildings become more important and the number of IoT devices used in various fields and applications continues to grow, semantic modeling based on ontology is needed for the standardization, generalization, reusability, and semantic enrichment of the systems, which contributes to the improvement of the system's efficiency [36], [37]. Ontology-based approaches in cryptography and access control enable the definition of semantically driven security policies, thereby enhancing the effectiveness of encryption models [38].

In this work, the ontology-based semantic method leads to inferencing implication and reasoning between fundamental IoT device properties in accordance with their practical utility and realizing a knowledge-based feature ensemble. This approach aims for the recognition of the semantic characteristics of IoT devices in smart buildings, dimensionality reduction of features through ontologies, and improved prediction accuracy of ensemble learning models. Hence, the proposed model could use the accuracy and error rates of an identification process through efficient identification of features for attribute-based encryption to choose features very accurately. We developed the proposed ontology using Protégé OWL and Resource Description Framework (RDF). The class diagram of the proposed ontology is shown in Fig. 5.

TABLE IV. The initial features of dataset.

Name	Date	Size	Role	Dep	Subject	TempS	LightS	MobilityS	QualityS	HumidityS	LightV	TempV	SN
AdaptiveBulb	2015	15	r_d1	r_dep	Relaxation	✕	✓□	✕	✕	✕	10	0	4
PersenceDetector	2020	30	r_a1	r_dep	Monitoring	✓□	✓□	✓□	✕	✕	0	23	8
HVAC	2018	12	r_l1	o_dep	Control	✓□	✓□	✓□	✓□	✓□	19	19	6
DataLogger	2017	10	r_l2	o_dep	Study	✓□	✕	✕	✓□	✓□	20	0	5
Lighting	2015	18	r_d2	o_dep	Checking	✕	✓□	✓□	✕	✕	16	0	4
Thermostat	2020	30	r_d3	r_dep	Meeting	✓□	✕	✕	✕	✓□	0	24	8
Library				o_dep								20	

TABLE V. The ontology-based semantic features dataset.

Category	Time stamp	Data volume	Role	Security management	Environmental control	Energy management	Sensor Ave
Social	2015-2020	15-30	d	✕	✓□	✕	10.6
Operational	2020-2020	30-30	a	✓□	✓□	✕	8
Detector	2017-2018	10-12	1	✕	✓□	✓□	5.5

TABLE VI. The definitive attributes list

Class	Semantic label
C1	General Service
C2	Control Service
C3	Alarm Trigger
C4	Archiving Trigger
C5	Disabling Trigger
C6	Portable
C7	None-Portable

TABLE VII. The value of swarm factors

Factors	Value
Population Size	40
Max Iteration	250
Constant ß, r1 & r2	1.5, [0.1]
Separation(s)	0.1
Cohesion (c)	0.7
Alignment(a)	0.1
Food & EnemyFactor (f) & (e)	1
Inertia Weight	0.9 - 0.2
Fitness	$kfoldLoss$
K- Fold	10
n, m	0.01, 0.99

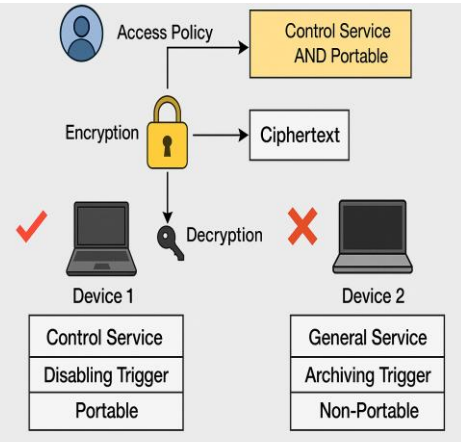

Fig. 4. The explanation of CP-ABE.

The *PhysicalObject* class represents a general category for any physical entity. The studied ontology comprises two subclasses: Platform and System. The Platform class defines an entity that can accommodate other entities, essentially serving as the installation or connection point for sensors and devices. In this study, 16 platforms are identified, including the bathroom, computer, dishwasher, living room, kitchen, oven, refrigerator, television, washing machine, four rooms, electricity, gas, and water.

The *System* class represents an abstract entity for infrastructure components and sensing mechanisms. A system may consist of multiple subsystems, each of which can also be classified as a system. The proposed smart home includes four major systems: home appliances, heating, lighting, and security systems.

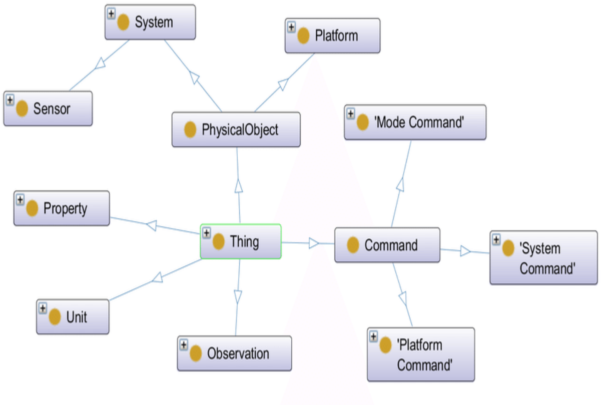

Fig. 5. The overall class diagram of the proposed ontology.

These classes and attributes incorporate the new data into the ontology, allowing for complex reasoning and inferencing with the Semantic Web Rule Language (SWRL). SWRL is the foundation for automated reasoning and for uncovering relationships between different entities in the ontology. They enable classification process and relationships between the concepts in the ontology.

Execution rules in the proposed ontology, are employed to perform specific actions. These rules are usually performed following an inference process and directly influence the hardware and software of the building. Examples of execution rules that trigger heating and cooling systems according to temperature readings, are presented in Algorithm 1. It receives sensor data and ontology-based service information as input and produces ontology-based commands as output.

The heating system rule states, if a room $(?r)$ has a sensor $(?s)$, and that sensor observes $(?o)$ a property $(?p)$ like temperature, then the respective value $(?v)$ is evaluated. If the temperature value $(?v)$ is less than or equal to 18 degrees Celsius $(swrlb\colon lessThan(?v, 18))$, a command $(?cmd)$ is sent to the system, indicating that the heating system should be activated $(activateHeatingSystem(?cmd))$. By applying this rule, it helps in automatic controlling of the heating system as soon as the temperature decreases from set point threshold or predefined threshold, making the smart home energy efficient & comfortable.

Algorithm 1. Sample of SWRL action rules

```
Input: Sensed Data, Ontology Info

Rull for activating Heating System
Room(?r) ^ hasSensor(?r, ?s) ^ hasObservation(?s, ?o) ^
hasProperty(?o, ?p) ^ value(?p, ?v) ^ swrlb:lessThan(?v, 18)
→ commands(?r, ?cmd) ^ SystemCommand(?cmd) ^
activateHeatingSystem(?cmd)

Rull for activating Cooling System
Sensor(?s) ^ hasObservation(?s, ?o) ^ hasProperty(?o, ?p) ^
value(?p, ?v) ^ swrlb:greaterThan(?v, 30) → commands(?s,
?cmd) ^ SystemCommand(?cmd) ^
activateCoolingSystem(?cmd)
Output: Action commands
```

Likewise, the cooling system rule states that if a sensor $(?s)$ value $(?v)$ exceeds 30 degrees Celsius $\left(swrlb\colon greaterThan(?v, 30)\right)$then a command $(activateCoolingSystem(?cmd))$ is issued to the cooling system to switch ON to cool down the temperature. When the temperature exceeds a threshold, this rule activates the cooling element, thus aiding environmental management processes, and increasing comfort in the smart house.

Feature extraction rules defined in the proposed ontology are utilized to decrease data dimensionality to improve classification model accuracy. These rules are defined based on relationships between entities and their attributes to extract the most significant features while eliminating less relevant ones. These rules are implemented via SWRL (Semantic Web Rule Language) which does not directly execute actions but rather allows the system to recognize new states and patterns within the smart home environment.

The proposed ontology includes some extraction rules of *SecurityManagementAbility* and *EnvironmentalControlAbility* features based on sensor attributes. Using this extracted feature by ontology-based approach helps reduce feature dimensionality and retain those features that are effective. This process is implemented in Algorithm 2. It takes sensor data and ontology-based information as input and generates ontology-driven semantic features as output to improve overall performance.

If the entity $(?d)$ executes an activity $(?a)$ with some numeric value $(?\mathrm{value}(?\mathrm{a}, ?\mathrm{activityValue}))$of the where $swrlb\colon greaterThan(?activityValue, 30)$, and if the entity is equipped with sensors $(?t)$and $(?m)$ that report meaningful values, $swrlb\colon greaterThan(?TempValue, 20)$ and $swrlb\colon greaterThan(?MobilityValue, 10)$, the answer to whether or not we can conclude that the entity (?d) has security management capabilities, which is defined as $hasSecurityManagement(?d, true)$. This rule is specifically giving security capability to those devices and entities with all three such features specified.

Algorithm 2. Sample of SWRL feature extraction

```
Input: Sensed Data, Ontology Info

Rull for extraxting Security Management
Device(?d) ^ Activity(?a) ^ belongsTo(?a, ?d) ^ value(?a, ?activityValue) ^ swrlb:greaterThan(?activityValue, 30) ^ Temp-sensor(?t) ^ belongsTo(?t, ?d) ^ value(?t, ? TempValue) ^ swrlb:greaterThan(?TempValue, 20) ^ Mobility-sensor(?m) ^ belongsTo(?m, ?d) ^ value(?t, ? MobilityValue) ^ swrlb:greaterThan(?MobilityValue, 10) → hasSecurityManagement (?d, true)

Rull for extraxting Environmental Control
Activity(?a) ^ value(?a, ?activityValue) ^ swrlb:greaterThan(?activityValue, 10) ^ (Temp-sensor(?t) v Humidity-sensor(?h) v Light-sensor(?l))→ hasEnvironmentalControl (?d, true)
Output: Semantic features
```

Activity $Activity(?a)$ is an abstract characteristic that is not directly observed by a dedicated sensor, but is employed to describe the state of the environment and the behavior of a user in the smart home. The data comes from the output of several sensors. For example, if a motion sensor is triggered and someone enters the room, $(?a, ?activityValue)$ would return a high value, e.g., 30, indicating high activity in the room. A light sensor being triggered in the same room, however, $(?a, ?activityValue)$ increases less significantly, reflecting a value of 10.

If an entity $(?d)$ performs an action $(?a)$ with $wrlb\colon greaterThan(?activityValue, 10)$ and has at least one of the sensors$(?h)$, $(?t)$, or $(?l)$, it is inferred that the entity $(?d)$ has environmental control capabilities, defined as *hasEnvironmentalControl(?d,true)*. Only the three sensors (temperature, humidity, and light) with a determined level of activity in this case are taken into account for developing and analyzing environmental control detection capabilities.

The extracted rules in the ontology studied here, which identify important sensor features, eliminate unnecessary features, and determine and remove dependent features, are shown in Algorithm 3.

If a sensor $(?s)$ has multiple observations $(?o)$ and a feature $(?p)$ exhibits significant variability that impacts the environment $isHighlyVariable(?p, true)$, that feature is selected as a key feature for subsequent processing $KeyFeature(?p)$. For instance, if temperature fluctuates considerably throughout the day, it is identified as a key feature.

Algorithm 3. Sample of SWRL feature extraction rules

```
Input: Sensed Data, Ontology Info

Rull for extraxting Key Features
Sensor(?s) ^ hasObservation(?s, ?o) ^ hasProperty(?o, ?p) ^ isHighlyVariable(?p, true) → KeyFeature(?p)

Rull for Removing Feature
Property(?p) ^ valueVariance(?p, ?v) ^ swrlb:lessThan(?v, 0.05) → RemoveFeature(?p)

Rull for Removing Correlated Feature
Property(?$p_1$) ^ Property(?$p_2$) ^ correlation(?$p_1$, ? $p_2$, ?cor) ^ swrlb:greaterThan(?cor, 0.9) → RemoveFeature(?$p_2$)
Output: Semantic features
```

If a sensor $(?s)$ has multiple observations $(?o)$ and a feature $(?p)$ shows extreme variance that affects the environment $isHighlyVariable(?p, true)$, then that feature is taken as a ket feature $KeyFeature(?p)$, and is carried for the next stage. For example, if a temperature varies a lot during the day, it is recognized as a a key feature.

On the other hand, if a feature exhibits minimal variation over time $valueVariance(?p, ?v)$and its variance is less than 0.05, it can be removed to save space. For instance, if the humidity varies less than 5% during the day, it is eliminated to lower the dimensionality of data.

Additionally, if two features $(?p1, ?p2)$ exhibit high correlation $correlation(?p_1, ?\ p_2, ?cor)$ and their correlation exceeds 0.9 $swrlb\colon greaterThan(?cor, 0.9)$, one of them can be eliminated $RemoveFeature(?p_2)$ to prevent redundancy. For instance, if temperature and power consumption in the heating system are highly correlated, one of them is removed to avoid redundant data.

Ultimately, the ontology-based approach in the proposed method extracts a set of concepts, relationships, rules, and semantic inferences through knowledge modeling within a specific domain, facilitating the extraction of basic IoT device features and their transformation into functional attributes to define a feature set based on domain knowledge.

Also, if two features $(?p1, ?p2)$ are highly correlated and their correlation value $(?p_1, ?\ p_2, ?cor)$ is more than 0.9 $swrlb\colon greaterThan(?cor, 0.9)$, one of them can be dropped $RemoveFeature(?p_2)$ to avoid duplication. For example, in case temperature and power consumption in heating system have high correlation, one of them is removed to not have duplicated data.

Finally, the ontology-based method of the proposed index allows to extraction of several concepts, relations, rules, and semantic inferences through knowledge modeling in a specific domain, this is useful for extracting simple IoT device characteristics and a way of copying them into functional characteristics to define a feature set as based knowledge in the task domain.

The ontology-driven semantic approach, combined with the Dragonfly algorithm, identifies significant semantic features while eliminating non-essential data. This procedure lowers the dimensionality of the attributes, which improves the function of learning models and attribute-based cryptography in smart homes.

4) The ensemble machine learning

The semantic ontology-based feature selection model has been evaluated utilizisng an ensemble machine learning algorithm that integrates multiple classifiers, such as GMDH, KNN, and SVM.

a) Group Method Data Handling

Inspired by nature, the Group Method of Data Handling (GMDH) is built on the complex structure of the human brain and is employed to mathematically model complex systems concisely and efficiently [39]. The GMDH is mainly associated with the structure implemented in a quadratic second-order transfer function (self-organizing feed-forward neural network) where regression approaches are used to estimate coefficients [40].

This algorithm proves to be highly effective due to its ability to generate accurate and efficient models for real-world problems, even when working with limited amounts of corrupted data. Moreover, the number of layers and neurons in the model is determined automatically, enabling the discovery of unbiased and objective models. GMDH automatically uncovers relationships between data and relevant input variables, and due to its sparse connectivity, it allows for rapid training [41].

The GMDH classifier is one of the significant base classifiers in method of ensemble learning of voting, because of its self-organizing structure, which ensures the optimality of model complexity and automatic selection of the most appropriate basic feature. It provides nonlinear fitting using polynomial, resulting in a more accurate prediction. More importantly, GMDH is robust to noise and high-dimensional datasets and can be adopted for real applications. Thanks to its adaptive learning process, it does not face the overfitting problem and provides a generalized model. Furthermore, GMDH is computationally efficient and can be used as part of an ensemble to enhance decision-making and total classification performance [41].

Root Mean Square Error ($RMSE$) is used as the objective function to determine the parameters of the regression for the power-law model [42]. The aim is to reduce the difference between the predicted outputs and the actual data labels. Equation 2 in this study defines the selection-pressure criterion that will be used to adjust precision. The user defines a, which has to between 0 and 1. The value of a is specified by the user and falls between 0 and 1. Neurons with $RMSE$ values lower than e_c (which is computed based on the strongest ($RMSE_{min}$) and weakest ($RMSE_{max}$) values of the neurons in the same layer) are excluded from further consideration.

$$e_c = a \times RMSE_{min} + (1 - a) \times RMSE_{max} \tag{2}$$

$$RMSE = \sqrt{MSE} \tag{3}$$

$$MSE = \frac{\sum_{i=1}^{n} (y_i - G_i)^2}{n} \tag{4}$$

The parameter effect has been investigated using trial-and-error techniques in several studies (e.g., [43], [44], [45]). Based on their findings, the optimal selection pressure is reported to be 70%, which is applied in the present study. The maximum number of layers has been suggested in the literature to be between 2 and 10 (e.g., [43], [44], [45]). Additionally, the number of neurons in prior studies (e.g., [43], [44]) has been set between 2 and 20, with a step size of 2. In the current study, we consider 9 layers and 6 neurons, with a selection pressure of 70%. It should be noted that the number of neurons may vary across layers. Therefore, the highest neuron count across all layers determines the maximum number of neurons, and we initiate the process with four neurons.

b) K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a commonly used classification algorithm in which each sample is classified by finding the K closest neighbors [19], [46]. The method for estimating distance typically depends on the specific goal and the nature of the data [8], [48]. We elected to use KNN as it is easy to implement, has high efficiency and is resistant to noisy training data. In this paper, proximity is measured using *Euclidean* distance, such that it was defined using Equation 5, where x represents the training data, x' refers to the test data, and n denotes the number of features.

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x_i')^2} \tag{5}$$

The choice of K significantly influences the algorithm, as the value of K defines the boundaries that separate the classes. We used 10-fold cross-validation on the training dataset to find the best K experimentally as part of the model building process, ensuring the chosen K generalizes well to new data. In this study, the value used is $k = 7$ neighbors because it yielded the best performance for KNN.

c) Support vector machine

Support Vector Machine (SVM) is a kernel approach for regression and classification that works very well. It is based on structural risk minimization in machine learning [21], [49]. For the Non-Linear data, Kernel trick is used to move the data point from its original feature space to a higher dimension space, where we can separate it linearly [50], [52]. Specifically, a set of samples is defined as $= \{(x_i, y_i)_{i=1}{}^{n} | x_i \in R^n, y_i = \{-1, 1\}, i = \{1, 2, 3, \ldots, n\}$, where x_i denotes the sample data, and y_i indicates the sample class. The hyperplane, which is what separates the sample, is what allows the samples to be separated by the largest margin and the objective function determines how to obtain this separation. The optimization problem is expressed as:

$$min \left\{\frac{1}{2} \|w\|^2 + c \sum_i \xi_i \right\}, \tag{6}$$

$$y_i(wx_i + b) \geq 1 - \xi_i \, c \geq 0, \qquad i = \{1, 2, 3, \ldots, n\}$$

The parameter c is the penalty coefficient.

The *Kernel*, c, and *Gamma* are the three crucial variables in the objective function [20], [45]. The *Kernel* specifies the hyperplane function type, where a common practice such as in this study, is to use the Radial Basis Function (*RBF*) kernel because of its good classification capabilities and applicability without prior knowledge of the dataset [47], [48], which is defined by the next equation:

$$K_{ebf}(X, X_i) = exp\left(-\frac{|X - X_i|^2}{\gamma^2}\right) \quad (7)$$

The parameter c serves as a penalty for classification errors incurred through creating decision boundaries using kernels [21], [50], [51], [52]. The hyper parameter *Gamma* determines the level of non-linearity in the RBF kernel. We used the GridSearchCV method to fine-tune these parameters, resulting in a value of $c = 0.1$ and *Gamma* ranging between 0.0001 and 10.

In this research, we selected SVM as it can overcome complex issues and provide accurate outcomes even with a small amount of data [21]. It has various advantages like it is immune to overfitting and uses different Kernel Tricks to fit complex relations between features and solve the non-linearity of data. The model stability and the ability to handle high-dimensional data is supported by the principle of hyperplane separation [54], [55].

d) The voting mechanism

We describe ensemble learning in this section and an approach that integrates multiple base classifiers using a majority voting scheme [25], [53]. Ensemble learning is the method where voting among different models occurs to get the final label of the data. The outcome with the maximum votes is the final [50], [51]. For classification, majority voting based ensemble learning helps to improve accuracy in the training stage by creating simpler models with high accuracy and less errors by combining outputs of different models [55].

Algorithm4. PseudoCode of ensemble Learning

```
Inputs:
$D_{tr}$: the training dataset
$D_{te}$: the testing dataset
EnsembleLearning (Dataset $D_{tr}$, Dataset $D_{te}$):
Class= {};
$M_{GMDH}$=CreateGMDHModel($D_{tr}$);
$M_{SVM}$=CreateSVMModel($D_{tr}$);
$M_{KNN}$=CreateKNNModel($D_{tr}$);
For i=1 To LEN($D_{te}$)
    $M_G[i] = M_{GMDH}$ ($D_{te}$[i]);
    $M_S[i] = M_{SVM}$ ($D_{te}$[i]);
    $M_k[i] = M_{KNN}$ ($D_{te}$[i]);
    C=MAX($M_G[i]$, $M_S[i]$ , $M_C[i]$);
    Class [i] = C;
End For
Output: Best Class
```

In this paper, we use the three algorithms: GMDH, KNN, and SVM. The motive for the mixture of these algorithms results from the respective dominant performance of all classifiers in the model generation. The architecture of the proposed ensemble learning model is illustrated in Fig. 7. The semantic feature set collected in the first stage is ingested as input, and training and evaluation datasets are generated to evaluate the model performance. Each base classifier fits its model on the training data independently. Then, each classifier is applied to predict the new labels for the data on the evaluation dataset. The outputs from all classifiers are combined using a majority-voting scheme, so the class with the highest number of votes is selected as the final prediction.

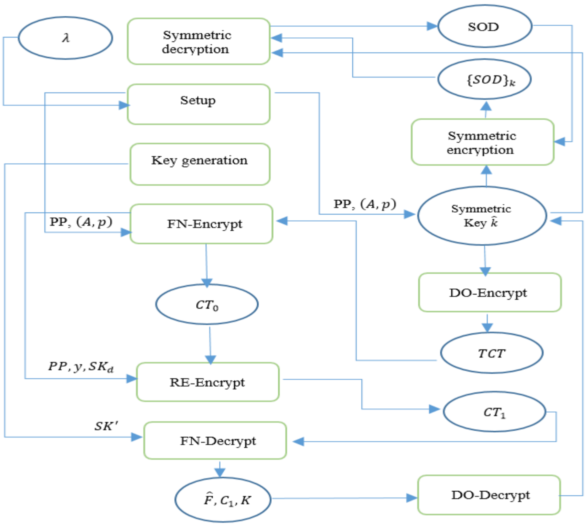

Fig. 6. The procedure of integrating AES with CP-ABE.

The proposed auto-trainable model operates independently, employing the semantic feature selection mechanisms available to the attribute authority. Consequently, the lengthy work for training and building relevant datasets, model generation, and model structure refinement are performed independently as part of the computing procedure, while the processing occurs in a high-power computing center. In this center, the class labels of the validation set are also predicted. Algorithm 4 shows the pseudo- code for the system that we proposed.

In this setup, D_{tr} denotes the training data, D_{te} represents the test data, M_{GMDH} corresponds to the GMDH classifier model, M_{SVM} stands for the SVM classifier model, and M_{knn} is the KNN the classifier model. The algorithm assigns the M_G class using the GMDH model, the M_S class using

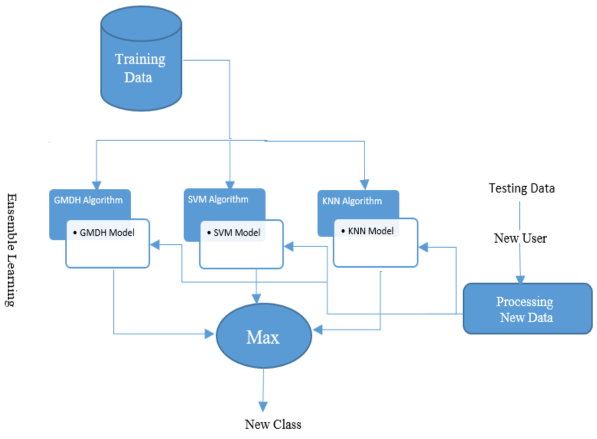

Fig. 7. The proposed ensemble learning machine.

the SVM algorithm, and the M_K class using the KNN algorithm. Finally, C represents the final predicted class, which is determined through majority voting.

5) The cryptosystem

This scheme makes use of symmetric and asymmetric encryption techniques [9]. The CP-ABE is employed for asymmetric encryption according to bilinear pairing operations over a BN elliptic curve with 128-bit security level. This technique is proven secure under the Decisional q-parallel BDHE assumption and supports LSSS access polices. Symmetric encryption is implemented using the AES algorithm and a 256-bit key, where plaintext is transformed into ciphertext. The following section provides details on how this works.

a) Ciphertext policy attribute-based encryption

This approach employs an outsourced CP-ABE scheme based on [9], which integrates proxy re-encryption to handle attribute revocation in the cloud data center. In this context, the attribute selection method based on the ensemble machine learning is used to reduce the number of attributes in different stages of the CP-ABE that impacts the efficiency and security of the CP-ABE. The attribute authority (*AA*) also gets the public key (*PK*) and master secret key (*MSK*), along with the attribute list of the IoT devices, to generate the users' secret key (SK_u). Attributes are an important part of the procedure itself (i.e., key generation and policy access definition) involved during the CP-ABE method, therefore optimization of semantic attributes improves the overall cryptographic structure of the proposed scheme.

As illustrated in Fig. 2, the *encryption* is performed in two stages and most of the encryption is offloaded to the fog. Initially, based on efficient AES encryption, the IoT device (*DO*) chooses a symmetric key (*K*) and an access policy. Once the data are sensed, the device encrypts this data using AES algorithm and symmetric key encrypted with partial ABE operation (*DO.PartialEncryption*) according to the access policy defined above. It sends this encrypted key and data towards the fog node (*FN*). The *FN* performs the key encryption in accordance with ABE (*FN.PartialEncryption*), encrypts the symmetric key based on the access policy, before uploading the information to the cloud service provider (*CSP*) along with the symmetrically encrypted information.

The *access policy* is given by the IoT device with respect to the optimized semantic attributes obtained by the ontology-based semantic module implemented in the attribute authority. The proposed semantic approach lightweightens and simplifies the access policy by select and reduces features. Since CP-ABE hides the access policy in the ciphertext, simplifying the policy reduces the size of the ciphertext and decreases the number and the complexity of encryption operations.

Decryption, similar to encryption, proceeds in two steps, with most of the decryption work offloaded to the fog. Here, the fog downloads the symmetrically encrypted ciphertext data and the ABE ciphertext (encrypted key *K*) stored in the *CSP* for the first. Then, the attribute-based decryption method is carried out (*FN.Decryption*) with a partial secret key. This gives us a collection of partially decrypted ciphertext and symmetrically encrypted data. The symmetric key is sent to the data user (*DU*), if the user's attributes comply with the defined access policy. The *DU* finishes decrypting by running the rest of the decryption process (*DO.PartialDecryption*), which it uses to retrieve the actual symmetric key and decrypt the data.

When attributes are revoked, the cloud service re-encrypts the data with new access structure by *proxy re-encryption* and the delegation key (K_d[1]), ensuring data security during the process of attribute revocation. This re-encryption method, which is a specialized variant of proxy re-encryption, relates to the fact that we decrypt relevant parts of the ciphertext and insert the corresponding new values under the data owner's identity. This sophisticated method enables users to delegate proxy access, enabling ciphertexts to be converted from one access structure to another [9], [56]. As this process is time-consuming and requires extensive computational power, the proposed policy considers a semantic features selection approach; thus, the access policy becomes less complex, readable, and less often updated. In addition, this scheme reduces the computational overhead of data owners and users and improves the security, accuracy, and efficiency of the model, which leads to lower encryption/decryption time, lower latency, and better service quality. The proposed CP-ABE scheme includes the *SETUP*, *KGEN*, *ENC*, *REENC*, and *DEC* algorithms.

Setup : In the beginning, the Setup function initializes the system by executing $Setup(\lambda) \rightarrow (PP, MSK, SK_d)$,

[1] Delegation Key

generating the public parameters *PP*, the master secret key *MSK*, and the delegation key SK_d. Following this, the Attribute Authority (*AA*) defines the semantic attribute space based on the security parameter λ, along with the set of possible attributes *S* derived from the semantic attribute set. The $BGGen(\lambda)$ function then constructs a bilinear group G with a prime order p and generator g. Subsequently, random values $a, \alpha_0, \alpha_1 \overset{R}{\leftarrow} \mathbb{Z}_p$ are selected, and the public parameter PP is determined as $PP = (g,\ g^a, \hat{e}(g,g)^{\alpha_0 + \alpha_1})$, where $\hat{y} = \hat{e}(g,g)$ represents the bilinear pairing. The master secret key MSK is defined as $MSK = (\alpha_0, \alpha_1,\ g^{\alpha_0 + \alpha_1})$, where $a = \alpha_0 + \alpha_1$, and the delegation key SK_d is given by $SK_d = g^{\alpha_1}$. Finally, the SK_d is sent to the Cloud Service Provider. The setup procedure is detailed in Algorithm 5.

KeyGen : When users receive an attribute set S from the semantic attributes list, the Attribute Authority arbitrarily chooses a value$c \overset{R}{\leftarrow} \mathbb{Z}_p$. It then runs the $KeyGen(PP, MSK, S) \rightarrow SK$ function to generate the secret key *SK*. The secret key is returned as $SK_u = \big(S, K, L, \bar{L},\ \{k_x\}_{x \in s}\big)$, where $K = g^{\alpha_0}.(\,g^a)^c = g^{\alpha_0 + \alpha_1}$, $L = g^c$, $\bar{L} = (\,g^a)^c = g^{ac}$ and $k_x = h(x)^c$ with $h \in \{0,1\}^* \rightarrow G$. Finally, SK_u is sent to the users, either *DO* or *DU*. The process of *KeyGen* is outlined in Algorithm 6.

Algorithm 5. *PseudoCode* of Setup

```
Inputs: $\lambda$
Algorithm: Set semantic attribute space
$BGGen(\lambda) \rightarrow (p, g, G, G_T, \hat{e})$
Pick random elements $a, \alpha_0, \alpha_1$ uniformly from $\mathbb{Z}_p$
Compute $PP = (g,\ g^a, \hat{e}(g,g)^{\alpha_0 + \alpha_1})$ where $\hat{y} = \hat{e}(g,g)$
Compute $MSK = (\alpha_0, \alpha_1,\ g^{\alpha_0 + \alpha_1})$ where $a = \alpha_0 + \alpha_1$
$SK_d = g^{\alpha_1}$
Send $SK_d$ to the $CSP$
Output: $PP, MSK, SK_d$
```

Algorithm 6. *PseudoCode* of KeyGen

```
Inputs: $PP, MSK, S$
Algorithm: Get a semantic attribute list of users (DO or DU)
Pick random element c uniformly from $\mathbb{Z}_p$
Compute $K = g^{\alpha_0}.(\,g^a)^c$ , $L = g^c$ , $\bar{L} = (\,g^a)^c$ , $k_x = h(x)^c$, $h \in \{0,1\}^* \rightarrow G$
Compute $SK_u = \big(S, K, L, \bar{L}\,, \{k_x\}_{x \in s}\big)$
Send $SK_u$ to the users
Output: SK
```

Encrypt: The encryption process in this scheme utilizes $Encrypt\big(\,PP, (A,p), \hat{k}\,\big) \rightarrow CT_0$ method and relies on the use of two procedures, *DO.Encrypt* and *FN.Encrypt,* which are utilized by the DO and FN components, respectively. Here, PP refers to the public parameter, and (A,p) represents the access policy, where A is a matrix of dimensions $l * n$ and $p: [l] \rightarrow U \subset \mathbb{Z}_p$. This setup is used to encrypt the message $\hat{k}$ and the data received from the Smart Objects Data (*SOD*). The Encrypt process, along with its two sub-modules, is outlined in Algorithm 7.

FN.Encrypt : Upon receiving *PP* and *TCT* as inputs, the final ciphertext CT_0 is generated by executing the $FN.Encrypt(\,PP,\ TCT\,) \rightarrow CT_0$ method, as outlined in Algorithm 8.

Algorithm 7. *PseudoCode* of Encrypt

```
Inputs: $PP, (A,p), \hat{k}$
Algorithm: Generate symmetric key $\hat{\mathrm{k}}$ from AES algorithm
Encrypt sensed data $\{SOD\}_k$ under symmetric key $\hat{k}$
Compute ciphertext $CT_0$ from symmetric key $\hat{\mathrm{k}}$ with access control $(A,p)$
Send $CT_0$, $\{SOD\}_k$ to $CSP$
Output: $CT_0$
```

Algorithm 8. *PseudoCode* of FN.Encrypt

```
Inputs: $PP,\ TCT$
Algorithm: Choose vector $\vec{v} = (s_0, v_2, \dots,\ v_n) \in \mathbb{Z}_p^n$
Calculate $l$ shares $\lambda_i = A_i.\overrightarrow{\mathrm{v}^t}$, i ∈ l.
Calc $C_2 = C_1\,.\,g^{s_0}, C_{i,1} = (\,g^a)^{\lambda_i}\,H(p(i))^{t_i}, C_{i,2} = g^{ti}, i \in [l]$
Compute $CT_0 = \Big((A,p), \hat{C}\,, C_1, C_2, \{\,C_{i,1}, C_{i,2}\}\Big)\,, i \in [l]$
Send $CT_0$, $\{SOD\}_{\hat{k}}$ to $CSP$
Output: $CT_0$
```

The fog node randomly picks a horizontal vector $\vec{v} \in \mathbb{Z}_p^n$ and determines the shares $\lambda_i = A_i.\overrightarrow{v^t}$ for $i \in l$, where $l \overset{R}{\leftarrow} \mathbb{Z}_p$. Subsequently, it computes the components of the ciphertext $C_2 = C_1\,.\,g^{s_0} = g^{s_0 + s}, C_{i,1} = (\,g^a)^{\lambda_i}\,H(p(i))^{t_i} = g^a\lambda, H(p(i))^{t_i}\,, C_{i,2} = g^{ti}, i \in \{1, 2, 3, \dots, l\}$. These represent sections of the source ciphertext. Finally, the fog node sends CT_0 and $\{SOD\}_{\hat{\mathrm{k}}}$ to the cloud provider.

DO.Encrypt : As illustrated in Algorithm 9, using the public parameter *PP* and the symmetric key $\hat{k}$, a intermediate ciphertext TCT is generated by executing $DO.Encrypt\big(\,PP, \hat{\mathrm{k}}\,\big) \rightarrow TCT$. The TCT is defined as $TCT = \Big((A,p), \hat{C}, C_1\Big)$, where $\hat{C} = \hat{k}.(\,y^a)^s = \hat{k}\,.\,y^{as}$ and $C_1 = \hat{k}\,.\,g^{s_0}\rrbracket$. Additionally, *DO* decrypts data obtained from smart objects using an efficient AES method. Subsequently, the *TCT* along with the encrypted data $\{SOD\}_k$ are sent to the fog nodes (*FN*).

ReEncrypt : When the cloud service provider (CSP) receives a delegation key $SK_d = g^{\alpha_1}$ where $\alpha_1 \overset{R}{\leftarrow} \mathbb{Z}_p$, a ciphertext CT_0, and an attribute y scheduled for revocation, it performs the re-encryption process by executing the method $ReEncrypt(\,PP, CT_0, y, SK_d) \rightarrow CT_1$, as described in Algorithm 10. The first step is to verify whether the attribute set in SK_d matches the access structure defined in

CT_0. If this condition holds, a re-encrypted ciphertext CT_1 is generated; otherwise, the output is "reject". Then, the *CSP* checks whether attribute y should be revoked from CT_1. If revocation is necessary, the CSP re-encrypts CT_0 to generate a new ciphertext CT_1 such that only users possessing attribute y are authorized to decrypt CT_1.

Algorithm 9. *PseudoCode* of DO.Encrypt

```
Inputs: $PP, \hat{k}$
Algorithm: Generate symmetric key $\hat{k}$ from AES
Choose a random element, $s$ at $\mathbb{Z}_p$
Compute $\hat{C} = \hat{k}.(y^a)^s$, $C_1 = \hat{k}.g^{s_0}$
Compute $TCT = \left((A,p), \hat{C}, C_1\right)$
Encrypt sensed data $\{SOD\}_{\hat{k}}$ under symmetric key $\hat{k}$
Send $TCT$, $\{SOD\}_{\hat{k}}$ to the FN
Output: $TCT$
```

Algorithm 10. *PseudoCode* of ReEncrypt

```
Inputs: $PP, CT_0, y, SK_d$
Algorithm: Control $\mathrm{SK_d}$ if its attributes satisfy $(A,p)$; otherwise, reject
Pick a random element v uniformly at $\mathbb{Z}_p$. $\mathrm{v} \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute v =bn256.RandomGT(rand.Reader)
Compute $D_1 = C_1^{1/v} = g^{s/v}, D_2 = (SK_d)^v = g^{a_1 v}$ Where $\alpha_1 \overset{R}{\leftarrow} \mathbb{Z}_p$ is a random element and $D_2$ is the ciphertext of the delegated key $SK_d$.
If y is a revoked attribute, pick a random key $\delta_y \in \mathbb{Z}_p$,
compute $C_{i,1} = C_{i,1}.H(p(i))^u$ , $\bar{C}_{i,2} = C_{i,2}.g^u = (g^{t_i+u})^{1/\delta_y}, i \in [l]$
Else pick a random integer $u \in \mathbb{Z}_p$, compute $\bar{C}_{i,1}$, $\bar{C}_{i,2} = C_{i,2}.g^u = g^{t_i+u}, i \in [l]$
Compute $CT_1 = \left((A,p), \hat{C}, C_1, C_2, D_1, D_2, \{C_{i,1}, C_{i,2}\}\right)$
Output: $CT_1$
```

Decrypt : The scheme executes $Decrypt(\, CT_1,\ SK_u\,) \rightarrow \hat{k}$ and utilizes two decryption processes, *DU.Decrypt* and *FN.Decrypt*, which are carried out by *FN* and *DU*, respectively, both of which have the necessary permissions for decryption. When provided with the secret key SK_u and the ciphertext CT_1 as inputs, these processes follow the steps outlined below to generate the symmetric decryption key $\hat{k}$.

Fn.Decrypt : In the scheme, the fog node (*FN*) executes the method $FN.Decrypt(\, CT_1,\ SK') \rightarrow \hat{F}$ by utilizing a portion of the secret key $SK' = (S, L, \bar{L}, \{k_x\}_{x \in s})$. This step performs a portion of the decryption algorithm. The algorithm identifies the set of indices $I = \{i \in [l] | \, p(i) \in S\}$ and computes the values $w_i \, A_i,\ i \in I$ for $\sum_{i \in I} w_i \, A_i = (1, 0, \dots, 0))$, where $\mathrm{A_i}$ is the *i*-th row of matrix *A*. Then, the intermediate ciphertext $\hat{F} \in G_{\mathrm{T}}$ is computed, and *FN* transmits $\hat{F}$ and *C1* to *DU*. The steps of *FN.Decrypt* are shown in Algorithm11.

DU.Decrypt : Using the method $DU.Decrypt(\, \hat{F}, C_1, K) \rightarrow \hat{K}$, the values$\hat{F}$, C_1, and $K \in G$ are received, and the plaintext is obtained through another part of the decryption process. The steps of *DU.Decrypt* are presented in Algorithm 12. The procedure of integrating AES with CP-ABE is illustrated in Fig. 6.

b) Advanced Encryption Standard

Advanced Encryption Standard (*AES*) is a symmetric block cipher commonly used for encrypting and decrypting data. Its simplicity, high efficiency, and low memory requirements, make it one of the most widely used encryption algorithms [2]. *AES* uses the same key for data encryption and decryption processes. The algorithm is known for its security, as its keys are relatively unfeasible for attackers to guess. Each *round* of AES occurs in parallel processing, whereas substitution and permutation are the two techniques used for the AES encryption. A round refers to the repeated cycles during the encryption and decryption procedures. AES is one of the most secure methods of encrypting systems. The presented framework includes IoT that act as data owners, the data is encrypted by AES with a 256-bit shared key, 14 rounds encrypts and decrypts the data [2], [5].

Algorithm 11 *PseudoCode of FN.Decrypt*

```
Inputs: $CT_1,\ SK'$
Algorithm: If the attribute set in $SK'$ does not satisfy the condition $(A,p)$ in $CT_1$, the process will be rejected. Otherwise, the set $I = \{i \in [l] | \, p(i) \in S\}$ is identified, and $\sum_{i \in I} w_i \, A_i = (1, 0, \dots, 0))$ is performed.
    Next, the value $\hat{F}$ is calculated as:
    Calculate $\hat{F} = \frac{\hat{C}.\prod_{i \in I} \hat{e}(\, C_{i,2}, K_{(p(i)}\,)^{wi} .\hat{e}(\, \bar{C}_2, \bar{L})}{\hat{e}\,(\prod_{i \in I}\, (\, C_{i,1}{}^{wi}, L)\, .\hat{e}(\, D_1, D_2)}$
    Send $\hat{F}$ and C1 are sent to DU for further processing.
Output: $\hat{F}$
```

Algorithm 12. *PseudoCode of DU. Decrypt*

```
Inputs: $\hat{F}, C_1,\ K$
Algorithm: Calculate $\hat{K} = \frac{\hat{F}}{\hat{e}(K, C_1)}$
    $\hat{K}$ is determined, the ciphertext is decrypted using this symmetric key.
Output: $\hat{K}$
```

VI. Results and discussion

This section begins by describing the simulation environment and the configuration of the proposed system, followed by an analysis of the evaluation metrics and criteria.

A. Experimental configurations

The evaluation was conducted on an Intel Core i7 machine with 16 GB RAM. The semantic feature selection method that was proposed was implemented and executed in MATLAB version 2018 running on Windows 10 OS. The ontology model was constructed with Protégé version 5.5 and based on the OWL2 data modeling language. The Semantic Web Rule Language (SWRL) was also utilized for rule-based logic. For the development and performance analysis of the cryptographic scheme, the Charm library version 0.43 was used on a VM having Ubuntu 18.04. Charm is an extensible framework for the rapid prototyping of cryptographic systems, with its implementation being in Python. To streamline development time and reduce code complexity, Python 3.4 and its components were employed. Table VIII outlines the configuration required for simulating the system, under which the presented model was deployed and its results examined.

B. Datasets

The evaluations were conducted utilizing five publicly available smart building datasets to validate the effectiveness of the presented approach. The datasets used include:

TABLE VIII. The configuration of simulating system.

Configuration	Desc
Operating system	Win 10 / Virtual machine using Ubuntu 18.04
RAM	16 Gigabytes
Processor	Intel Core i7 processor
Programming tools	Protégé ۵/۵ with OWL2 & SWRL / Matlab 2018 / Version 0.43 of the Charm library, utilizing Python 3.4

- ***HomeC dataset:*** This dataset features weather information specifically for a smart home, providing insights into how external weather conditions may impact energy consumption and building performance [57].

- ***SBS dataset:*** Comprising five distinct types of time-series data collected from various sensors, this dataset was gathered over a week at the University of California, Berkeley, allowing for a comprehensive analysis of sensor data in a smart building context [58].

- ***OSH dataset:*** This dataset focuses on modular building design and modeling, particularly in intelligent homes, offering a resource for evaluating design choices and their implications on energy efficiency [59].

- ***CU-BEMS dataset:*** Featuring electrical measurements and environmental data from a commercial building in Thailand, this dataset is useful for understanding energy usage patterns and environmental conditions in a real-world setting [60].

- ***IEQ dataset:*** This dataset provides data on both static and dynamic power consumption in smart homes, including measurements from sensors and actuators, which can help assess indoor environmental quality and energy efficiency [61].

C. Proof of Security

As stated in [9], if the BDHE problem is assumed to be difficult, our cryptographic system can be proven to be securely protected under the proposed security model, as stated in the subsequent theorem:

Theorem: If the decisional q-parallel BDHE assumption holds, then no polynomial-time adversary can selectively break our system with a challenging matrix of order $m * n$, $m, n <= q$.

Consider a polynomial-time adversary $\mathcal{A}$ who is capable of breaking our system with a non-negligible advantage $Adv_{\mathrm{A}}^{q}(\lambda)$ in the selective security game. This adversary could be utilized to build a simulator that breaks the assumption with non-negligible probability. This leads to a contradiction regarding the validity of the decisional q-parallel BDHE assumption, thus completing the proof.

Initialization: Initially, the challenger C runs the $BGGen(\lambda) \rightarrow (p, g, G, G_T, \hat{e})$ algorithm to construct a bilinear group based on the security parameter λ. The adversary A then sends the challenge access policy (A, p) along with the matrix A^* of order $m * n$, where $m, n <= q$, and a list of revoked attributes R_y, which contains all the revoked user attributes, to the challenger C.

Setup: The challenger C executes the Setup process to establish the public parameters *PP*, the delegation secret key SK_d, and the master key *MK*. C then sends the public parameters *PP* and SK_d to adversary A and the cloud service provider (*CSP*).

Phase 1: The adversary A makes several queries for the retrieval of private keys based on each identifier and attribute set $(ID_i,\ S_i)$,where $1 \leq i \leq q$, ensuring that the attribute sets are not part of the access policy. The challenger C generates the private key SK_{ui} and sends it to the adversary A. None of the attribute sets S_i in the queries match the access structure.

Challenge: The adversary A sends a message $M \in \{0,1\}$ to the challenger C. The challenger C randomly selects $\mathrm{b} \in \{0,1\}$ and performs the operations $Encrypt(\ PP, A^*, \widehat{\mathrm{M}_b}\ ) \rightarrow\ CT_0^{(b)}$ and $ReEncrypt(\ PP, CT_0^{(b)}, y^*, \mathrm{SK_d}) \rightarrow\ CT_1^{(b)}$. C then sends the final ciphertext $CT_1^{(b)}$ to adversary A.

Phase 2: The adversary A submits more queries for the retrieval of private keys, and the responses are the same as

in *Phase 1*, meaning none of the attribute sets S_i are allowed in A^*.

Guess: The adversary A makes a guess b' for b. If $b'= b$, the adversary wins the game, and the challenger C returns 0 to estimate the value $T =\ \hat{e}(g, g)^{sa^{q+1}}$. Thus, the simulation is perfect, and we have $Pr\left[A(\ \vec{y}, \hat{e}(g, g)^{sa^{q+1}}\ ) = 0\ \right] = Adv_{\mathrm{A}}^{q}(\lambda) +\ ½$. Otherwise, T is treated as a random element of group. In this case, with a random group R, we have $Pr\left[A\left(\vec{y}, R \overset{R}{\leftarrow} G_T\right) = 0\right] =\ ½$, indicating that b is independent of the adversary's perspective. Consequently, the adversary is able to break the decisional q-parallel BDHE assumption with a non-negligible advantage, which contradicts the assumption.

$$A\mathrm{dv}_{\mathrm{A}}^{q}(\lambda) = Pr\left[A\left(\vec{y}, \hat{e}(g,g)^{sa^{q+1}}\ \right) = 0\ \right] \\ -\ \Pr\left[A\left(\vec{y}, R \overset{R}{\leftarrow} G_T\right) = 0\right].$$

D. Performance evaluation

We provide the outcomes of our experiments and conduct a comparative analysis with alternative methods. The proposed model, which incorporates semantic feature selection based on ontology along with ensemble classification, was tested on different datasets and evaluated using various metrics, including accuracy, precision, specificity, recall, error rate, and F1-score. These evaluations, based on the formulas provided in Table IX, were performed for multi-class classification on the HomeC, OSH, SBS, IEQ, and CU-BEMS data collections to assess the model's performance in addressing the problem.

Table IX details the evaluation formulas for each metric used in this study, providing a clear framework for understanding the effectiveness of our model. In this context, there are c classes, and we observe TP_i(True Positive) as the count of actual positives for class i, FP_i (False Positive) as the count of incorrect positives for class i, TN_i (True Negative) as the count of actual negatives for class i, and FN_i (False Negative) as the count of incorrect negatives for class i. These values are extracted using the confusion matrix, which summarizes the classification predictions made on the test datasets [62].

The classification performance evaluation metrics, computed by the proposed model and other algorithms like LSTM[2], RDNN[3], GBT[4], KNN, SVM, and RF[5], are presented for the multi-class classification of the HomeC dataset in Table X, based on the relations outlined in Table IX. Table XI presents a comparison of the proposed model's performance with other classification methods on the SBS dataset, while Table XII shows the results of the proposed model's evaluation against alternative approaches applied to the OSH data collection. Lastly, Tables XIII and XIV present the performance comparison of the presented model with other models on the CU-BEMS and IEQ datasets, respectively, using metrics such as accuracy, error rate, precision, specificity, recall, and F1-Score for multi-class data.

A review of Tables X to XIV shows that the proposed classification model outperforms others in terms of accuracy across five datasets. This superiority is derived from the implementation of a semantic-based ontology approach, optimized feature selection, and ensemble learning. Moreover, in all of the smart building datasets, the proposed model outperforms with better precision and recall and specificity values. The enhanced performance is attributed to the ontology-based semantic modeling and the refined feature selection. Furthermore, the F1-Score metrics for the suggested model in this work show consistently high values in all the data collections, indicating its superior overall performance. The proposed model structure is simple and clear, so the error rate in feature selection is small, and a more accurate and effective model can be obtained. In addition, some existing models like LSTM and RDNN with large parameters are sensitive to the training data or noisy data and can easily cause overfitting when the data is insufficient. Also, their accuracy decreases on limited training data or non-optimized parameters, while the proposed voting-based ensemble model built by SVM, KNN, and GMDH can avoid this situation. Models such as SVM, KNN, or GMDH could be less prone to overfitting in low-volume or noisy data due to their structural and operational simplicity. They can help avoid unreasonable complexity with the right settings. It also prevents overfitting by combining the predictions of the ensemble model, including SVM, KNN, and GMDH. This estimation diversity gives errors from one model a chance to be corrected elsewhere, making them less sensitive to noisy or outlier datasets. Also, these three models are combined to reduce noise, improving the model's overall accuracy. Furthermore, our semantic ontology-based feature selection aims to avoid overfitting by removing irrelevant, noisy, and redundant features, increasing model performance and complexity, and improving the model to stratify more generalized. The proposed model lessens error rates, enhances F-score, and maintains a balance between sensitivity and specificity, which is crucial in the case of imbalanced datasets. As well as, the training process of some existing models like LSTM and RDNN involving Backpropagation Through Time (BPTT) requires high memory and computational time but our work executes lighter and faster models, leading to less execution time and lower error rates.

This study integrates the key semantic features extracted from IoT devices in intelligent building systems with the cryptographic process Furthermore, the cryptographic computation time of the proposed model is compared with the approach introduced in [9] over the five datasets used on smart building, namely, OSH, IEQ, SBS, HomeC, and CU-BEMS.

TABLE IX. Mathematical formulations for evaluation and definitions of multi-class metrics

The approach presented in [9] relies on basic, non-performance-based features and has a slow inflation of attributes set that may restrict compatibility with resource-limited devices and fog environments. This increases the size of ciphertext, length of secret key, and time of cryptographic processing that adversely affect security and user privacy. This paper proposes a refined version of the CP-ABE

[۲] Long Short-Term Memory

[۳] Recurrent Deep Neural Network

[٤] Gradient-Boosted Trees

[٥] Random Forest

Metrics	Specs	Evaluation focus
Accuracy	$\frac{\sum_{i=1}^{n}\frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i}}{n}$	The average of the correctly classified instances relative to the total instances for each class. It is important to note that a model performs optimally when it achieves a higher accuracy rate.
Precision	$\frac{\sum_{i=1}^{n}\frac{\mathrm{TP}_i}{\mathrm{TP}_i+\mathrm{FP}_i}}{n}$	The average of true positive instances correctly predicted from all actual positive instances for each class.
Recall	$\frac{\sum_{i=1}^{n}\frac{\mathrm{TP}_i}{\mathrm{TP}_i+\mathrm{FN}_i}}{n}$	The average of the instances predicted as positive that are actually positive, out of all instances predicted as positive by the model, for each class.
Specificity	$\frac{\sum_{i=1}^{n}\frac{TN_i}{TN_i+FP_i}}{n}$	The average of the true negative instances predicted by the model out of all actual negative instances predicted as negative for each class.
Error	$\frac{\sum_{i=1}^{n}\frac{FP_i+FN_i}{TP_i+TN_i+FP_i+FN_i}}{n}$	The ratio of the correctly predicted instances to the total number of instances for each class.
F1-Score	$\frac{2*Precision_m*Recall_m}{Precision_m+Recall_m}$	The harmonic mean, derived from false negatives, false positives, and sensitivity which is calculated for each class. The F1-score may provide a more meaningful evaluation metric than accuracy, especially in cases of imbalanced class distributions.

scheme in [9] by using a semantic ontology and ensemble learning model that focus on attribute reduction and performance optimization, thus, reducing security concerns.

Fig. 8 illustrates the encryption computation time for the proposed method and the method from [9] on the five data collections. Fig. 9 shows the decryption time results for both methods. The proposed approach outperforms the comparison method in terms of minimizing the calculation time of encryption and decryption, saving computation time for data owners, users, and fog nodes. The mean reduction in encryption and decryption times is 2.99 ms and 2.86 ms, respectively, compared to the method in [9]. The average decrement in encryption and decryption times is 2.99 ms and 2.86 ms, respectively, over the method mentioned in [9]. The key reasons are the adoption of optimal features selected by the Dragonfly optimization algorithm and the semantic approach, which eliminates irrelevant features and focuses on the most important and related ones.

The attribute revocation time is another important assessment metric for the proposed model, which consists of during the exclusion of attributes, update of the key and access structure, and the re-encryption of data. As shown in Table XV, the proposed model shows a clear advantage over the method in [9], with an average revocation time improvement of 18.6 ms. The revocation time is relied upon reconstructing access policies, re-encrypting data, and regenerating secret keys and therefore, makes the proposed scheme superior and quicker in comparison with the method in [9] due to the use of meaningful and influential features.

Finally, Table XVI compares the storage requirements for generating the private key and defining the access structure on the smart building data collections (HomeC, OSH, SBS, IEQ, and CU-BEMS) between the proposed method and the scheme in [9]. According to Table XVI, average memory consumption of constructing private key is reduced by 13.56 KB, and space overhead of access structure is optimized by 10.4 KB. This reduction in storage is attributed to the semantic ontology-based model and the Dragonfly optimization algorithm which minimize the number of attributes included in the access policy and the process of private key generation. As a result the data structure becomes very small as there is no need to save unnecessary information.

The key and access structure's smaller size leads not just to efficient use of system resources but also increased complexity of data processing. Furthermore, since the access structure is included in the cipher text, the improvement of access policies and their simplification is essential to increase the overall performance of attribute-based cryptographic schemes. Finally, the use of ontological approaches offers more flexibility and compatibility for changes to data and policies, contributing also to the scalability of the system in terms of the volume of data and new complexities, while preserving high levels of security and performance.

Our ontology-based model proposed in this paper aims to tackle one of the biggest challenges in smart environments with diverse IoT devices, namely scalability and heterogeneous IoT data management. Combining both semantic reasoning and ensemble-based learning, this framework builds a systematic model for deriving essential characteristics from heterogeneous data sources to produce Key and control access to the same through abstracting the semantic links between data, devices, and humans. Also, feature selection is important in improving cryptographic operations by removing irrelevant or noisy attributes, and in turn lessens computation complexity and improves system performance when dealing with large amounts of data.

Moreover, ontology integration and feature selection algorithms further validate the flexibility of access structures that facilitate adaptive security policy deployment in changing settings. Such adaptability is crucial for big data management and dynamic access control in the IoT ecosystem due to the variety of devices, the continuous changes of data conditions, and the increasing number of users, which require scalable and flexible cryptographic models. One of the key purposes of the presented model is to improve context-aware security policy reasoning of the big data processing node and optimize the CP-ABE process of the big data processing node to better enhance the security while still being able to maintain good scalability and efficiency to deal with those large-scale and dynamic data. Thus, this model is an intelligent, scalable, efficient cloud-fog-IoT computing model that can efficiently process complex tasks with high data volumes and dynamic access conditions, which facilitates the efficiency from the cloud to the fog and IoT environments.

VII. Conclusion and future work

This study introduces a novel cooperative approach designed to bolster the security of smart buildings within fog computing environments. The approach emphasizes optimal feature selection and accurately applying relevant attributes in ciphertext-policy attribute-based encryption (CP-ABE), while also enhancing the performance of encryption, decryption, and feature revocation. To accomplish these objectives, a semantic-based method utilizing an ontology framework is employed, in conjunction with the Dragonfly optimization algorithm and a KNN kernel, to effectively identify key features. Furthermore, the ontology-based strategy, paired with inference rules, facilitates the reduction of data dimensionality and transforms the data into meaningful values. This comprehensive methodology not only improves security but also ensures that the system is efficient and responsive to the dynamic needs of smart building environments.

The final stage of feature extraction and selection was performed using ensemble learning, utilizing a combination of GMDH, SVM, and KNN algorithms. This study used ciphertext-policy attribute-based encryption (CP-ABE) integrated with the AES algorithm for the encryption, decryption, and feature revocation processes on the reduced semantic features. The evaluation of the proposed model was conducted on five datasets, OSH, HomeC, SBS, IEQ, and CU-BEMS and the results illustrated that our approach effectively identifies key features and surpasses existing methods in performance. A comparative analysis with other techniques, such as LSTM, RDNN, KNN, GBT, SVM, and RF, demonstrated superior outcomes in various metrics, including accuracy, precision, specificity, error rate, recall, and F1-score. Moreover, the findings indicate that outsourcing encryption and decryption tasks to fog nodes significantly reduces computational costs. The result of evaluating the proposed model on five smart building datasets indicates that our approach is able to successfully identify important features and outperform existing methods in terms of performance. Performances were superior to LSTM, RDNN, KNN, GBT, SVM, and RF under different metrics (accuracy, specificity, precision, error rate, recall, and F1-score). In addition, outsourcing the encryption and decryption tasks at fog nodes significantly reduced the computation cost. Adopting a semantic-based approach to feature selection for encryption and decryption further improved efficiency in execution times, achieving 2.99 ms encryption and 2.86 ms decryption compared to traditional methods. It was also observed that the time to revoke the feature was reduced to 18.6 ms compared to the previous methods. The study also showed a 13.56 KB and a 10.4 KB reduction in memory consumption during the construction of the private key and the access policies, respectively. The main reason for these improvements is with the optimized feature selection and the semantic ontology-based model that simplifies the complexity of data structures. The proof of security of the proposed approach was shown in the standard model with respect to in memory consumption during the construction of the private key and the access policies, the DBDH assumption. Nevertheless, these features provide efficiency and flexibility for resource-constrained platforms in fog-based smart buildings.

The findings of the study represent major improvements in cryptographic computations and data classification in smart homes, however, because the core of methodology is not implemented on the IoT devices and resource-constrained fog nodes, it should be regarded as a relevant limitation. Moreover, other key components such as transmission delays and energy costs required for data exchange, which impact the overall performance of the system, are not covered in the study. Future works will look into other optimization methods like genetic, honey bee, advanced cat, and ant colony optimization as an alternative of the its Dragonfly to produce a salient features and refine the model.

This work also needs to be expanded with efficient policy updates in heterogeneous IoT environments, accommodating diverse device attributes, constant data flows, and changing security policies in heterogeneous IoT networks. Lightweight mechanisms, such as incremental re-encryption, may facilitate policy adaptation with lower computational costs. This would enable scalable policy enforcement with quantization of system latency under network heterogeneity. Such improvements would facilitate a robust and adaptable access control paradigm, providing secure and effective policy administration in extensive and changing IoT environments.

In addition, our future research will also include the application of deep bidirectional and reinforcement neural networks, in contrast to the ensemble learning method for the identification of features and classification of data, which may also improve the effectiveness of the proposed method. Another interesting avenue to explore would be to use fuzzy logic instead of the semantic ontology based approach to benefit better results. Lastly, a challenge is an interesting direction for future research, which is to embed our proposed model within lattice-based ABE schemes where constructions are highly complex and requires high storage as well as time consumption.

SBS Data Collection

Metrics %	RF	SVM	GBT	KNN	RDNN	LSTM	Proposed work
Accuracy	98.08	97.51	92.81	93.82	94.44	89.34	**99.01**
Precision	95.8	94.95	91.16	91.57	92.81	88.61	**97.91**
Recall	97.25	97.11	97.63	97.02	97.81	98.22	**98.60**
Specificity	98.09	96.92	92.89	92.71	94.31	88.12	**99.03**
Error	1.92	2.49	7.19	6.18	5.56	10.66	**0.99**
F1-Score	96.93	96.21	91.98	92.68	93.62	88.97	**98.46**

TABLE XI. Performance metrics on SBS.

CU-BEMS Data

Metrics %	RF	SVM	GBT	KNN	RDNN	LSTM	Proposed work
Accuracy	98.54	97.8	94.44	94.76	92.44	90.96	**98.88**
Precision	96	95.12	92.64	92.34	90.94	89.94	**97.2**
Recall	97.32	97.09	97.72	97.1	97.89	98.3	**98.6**
Specificity	98.32	97.01	94.32	94.87	92.01	90.98	**99.91**
Error	1.46	2.2	5.56	5.24	7.56	9.04	**1.12**
F1-Score	97.25	96.44	93.53	93.53	91.68	90.45	**98.03**

TABLE XIII**.** Performance metrics on CU-BEMS.

	Data Collection					
millisecond	HomeC	OSH	SBS	IEQ	CU-BEMS	AVG
Proposed Work	108	68	92	17	39	64.8
The Scheme[9]	135	90	110	27	55	83.4

TABLE XV. Comparative evaluation of attribute revocation time in the proposed approach with the scheme in [9].

HomeC Data Collection

Metrics %	RF	SVM	GBT	KNN	RDNN	LSTM	Proposed work
Accuracy	98.96	98.43	95.06	95.79	92.86	91.59	**99.21**
Precision	96.2	95.7	93.19	93.28	91.29	90.46	**98.31**
Recall	97.14	97.11	97.73	97.12	97.9	98.31	**98.50**
Specificity	98.23	98.76	95.08	95.03	92.7	91.61	**99.27**
Error	1.04	1.57	4.94	4.21	7.14	8.41	**0.79**
F1-Score	96.67	97.05	94.12	94.52	92.07	91.02	**98.76**

TABLE X. Performance metrics on HomeC.

OSH Data Collection

Metrics %	RF	SVM	GBT	KNN	RDNN	LSTM	Proposed work
Accuracy	98.08	97.5	92.77	93.82	94.41	89.31	**98.94**
Precision	95.8	94.95	91.16	91.57	92.81	88.61	**97.23**
Recall	97.45	97.1	97.62	97.02	97.8	98.22	**98.6**
Specificity	98.01	97.6	92.61	98.71	94.5	89.4	**99.95**
Error	1.92	2.5	7.23	6.18	5.59	10.69	**1.06**
F1-Score	96.93	96.21	91.96	92.68	93.60	88.96	**98.08**

TABLE XII. Performance metrics on OSH.

IEQ Data Collection

Metrics %	RF	SVM	GBT	KNN	RDNN	LSTM	Proposed work
Accuracy	98.06	97.46	92.65	93.68	94.32	89.13	**98.2**
Precision	95	94.87	91.03	91.41	92.69	88.44	**96.2**
Recall	96.6	97.05	97.58	96.96	97.76	98.19	**98.84**
Specificity	98.01	97.32	92.6	93.69	94.12	89.2	**99.91**
Error	1.94	2.54	7.35	6.32	5.68	10.87	**1.80**
F1-Score	96.51	96.15	91.83	92.53	93.50	88.78	**97.19**

TABLE XIV. Performance metrics on IEQ.

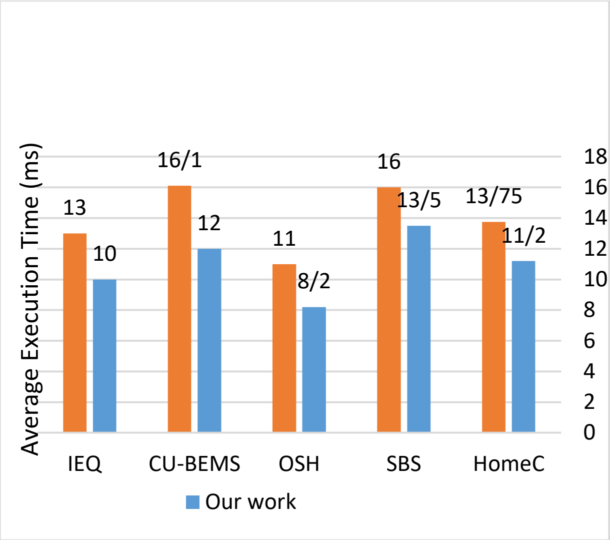

Fig. 8. Comparison of encryption execution time.

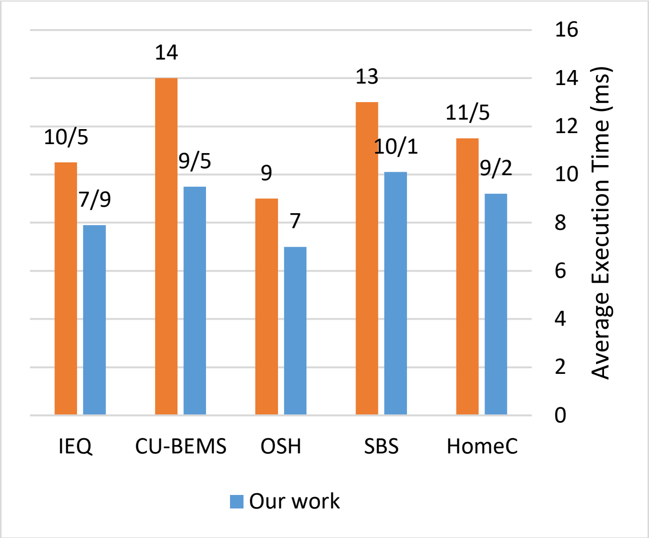

Fig. 9. Comparison of decryption execution time.

TABLE XVI. Comparison of private key storage and access structure requirements in the proposed method vs. [9].

	Access Policy (Kbytes)					**Key Generation (Kbytes)**				
Data Collection	HomeC	OSH	SBS	IEQ	CU-BEMS	HomeC	OSH	SBS	IEQ	CU-BEMS
Proposed Work	20	20	17	17.5	17	14.2	16	12	12	11
The Scheme [9]	30	31	27	28	27.5	28	29	26	26	25

REFERENCES

[1] Miraz, M.H., Ali, M., Excell, P.S. and Picking, R., 2015. A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT). 2015 Internet Technologies and Applications (ITA), pp.219-224. doi: 10.1109/ITA.2015.7317398.

[2] Rasori, M., La Manna, M., Perazzo, P. and Dini, G., 2022. A survey on attribute-based encryption schemes suitable for the internet of things. IEEE Internet of Things Journal, 9(11), pp.8269-8290. doi: 10.1109/JIOT.2021.3122685.

[3] Li, H. and Jing, T., 2020. A ciphertext-policy attribute-based encryption scheme with public verification for an IoT-fog-cloud architecture. Procedia Computer Science, 174, pp.243-251. doi: 10.1016/j.procs.2020.06.089.

[4] Sethi, K., Pradhan, A. and Bera, P., 2021. PMTER-ABE: a practical multi-authority CP-ABE with traceability, revocation and outsourcing decryption for secure access control in cloud systems. Cluster Computing, 24, pp.1525-1550. doi: 10.1007/s10586-020-03184-w.

[5] Khalid, T., Abbasi, M.A.K., Zuraiz, M., Khan, A.N., Ali, M., Ahmad, R.W., Rodrigues, J.J. and Aslam, M., 2021. A survey on privacy and access control schemes in fog computing. International Journal of Communication Systems, 34(2), p.e4181. doi: 10.1002/dac.4181.

[6] Asghari, P., Rahmani, A.M. and Javadi, H.H.S., 2019. Internet of Things applications: A systematic review. Computer Networks, 148, pp.241-261. doi: 10.1016/j.comnet.2018.12.008.

[7] Sarma, R., Kumar, C. and Barbhuiya, F.A., 2022. MACFI: A multi-authority access control scheme with efficient ciphertext and secret key size for fog-enhanced IoT. Journal of Systems Architecture, 123, p.102347. doi: 10.1016/j.sysarc.2021.102347.

[8] Wen, M., Chen, S., Lu, R., Li, B. and Chen, S., 2019. Security and efficiency enhanced revocable access control for fog-based smart grid system. IEEE Access, 7, pp.137968-137981. doi: 10.1109/ACCESS.2019.2942601.

[9] Zhao, J., Zeng, P. and Choo, K.K.R., 2021. An efficient access control scheme with outsourcing and attribute revocation for fog-enabled E-health. IEEE Access, 9, pp.13789-13799. doi: 10.1109/ACCESS.2021.3051098.

[10] Kazemi, M., Ghanbari, S. and Kazemi, M., 2020. Divisible load framework and close form for scheduling in fog computing systems. Recent Advances on Soft Computing and Data Mining: Proceedings of the Fourth International Conference on Soft Computing and Data Mining (SCDM 2020), Melaka, Malaysia, January 22–23, 2020, pp.323-333. Springer International Publishing. doi: 10.1007/978-3-030-36056-6_30.

[11] Saidi, A., Nouali, O. and Amira, A., 2022. SHARE-ABE: an efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and Fog computing. Cluster Computing, 25(1), pp.167-185. doi: 10.1007/s10586-021-03383-5.

[12] Luo, F., Al-Kuwari, S., Wang, F. and Chen, K., 2021. Attribute-based proxy re-encryption from standard lattices. Theoretical Computer Science, 865, pp.52-62. doi: 10.1016/j.tcs.2021.04.014.

[13] Wu, Z., Shi, R.H., Li, K. and Yang, Y., 2022. Attribute-based data access control scheme with secure revocation in fog computing for smart grid. Cluster Computing, 25(6), pp.3899-3913. doi: 10.1007/s10586-022-03560-1.

[14] Xin, X., Yang, Q. and Li, F., 2020. Quantum public-key signature scheme based on asymmetric quantum encryption with trapdoor information. Quantum Information Processing, 19(8), p.233. doi: 10.1007/s11128-020-02686-2.

[15] Tu, S., Waqas, M., Huang, F., Abbas, G. and Abbas, Z.H., 2021. A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing. Computer Networks, 195, p.108196. doi: 10.1016/j.comnet.2021.108196.

[16] Sarma, R., Kumar, C. and Barbhuiya, F.A., 2022. MACFI: A multi-authority access control scheme with efficient ciphertext and secret key size for fog-enhanced IoT. Journal of Systems Architecture, 123, p.102347. doi: 10.1016/j.sysarc.2021.102347.

[17] Fu, X., Ding, Y., Li, H., Ning, J., Wu, T. and Li, F., 2022. A survey of lattice based expressive attribute based encryption. Computer Science Review, 43, p.100438. doi: 10.1016/j.cosrev.2021.100438.

[18] Dixit, S., Joshi, K.P., Choi, S.G. and Elluri, L., 2022, May. Semantically rich access control in cloud ehr systems based on ma-abe. In 2022 IEEE 8th intl conference on big data security on cloud (BigDataSecurity), IEEE intl conference on high performance and smart computing, (HPSC) and IEEE intl conference on intelligent data and security (IDS) (pp. 1-10). IEEE. doi: 10.1109/BigDataSecurity53610.2022.00004.

[19] Maleki, N., Zeinali, Y. and Niaki, S.T.A., 2021. A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. Expert Systems with Applications, 164, p.113981. doi: 10.1016/j.eswa.2020.113981.

[20] Hazman, C., Benkirane, S., Guezzaz, A., Azrour, M. and Abdedaime, M., 2022, November. Intrusion detection framework for IoT-based smart environments security. In The International Conference on Artificial Intelligence and Smart Environment (pp. 546-552). Cham: Springer International Publishing. doi: 10.1007/978-3-030-97343-2_55.

[21] Panthi, M. and Das, T.K., 2022. Intelligent intrusion detection scheme for smart power-grid using optimized ensemble learning on selected features. International Journal of Critical Infrastructure Protection, 39, p.100567. doi: 10.1016/j.ijcip.2022.100567.

[22] Rahman, H. and Hussain, M.I., 2021. A light-weight dynamic ontology for Internet of Things using machine learning technique. ICT Express, 7(3), pp.355-360. doi: 10.1016/j.icte.2021.02.002.

[23] Saba, D., Sahli, Y. and Hadidi, A., 2021. An ontology based energy management for smart home. Sustainable Computing: Informatics and Systems, 31, p.100591. doi: 10.1016/j.suscom.2020.100591.

[24] Khan, P.W. and Byun, Y.C., 2020. Genetic algorithm based optimized feature engineering and hybrid machine learning for effective energy consumption prediction. IEEE Access, 8, pp.196274-196286. doi: 10.1109/ACCESS.2020.3039826.

[25] Tasci, E., 2020. Voting combinations-based ensemble of fine-tuned convolutional neural networks for food image recognition. Multimedia Tools and Applications, 79(41), pp.30397-30418. doi: 10.1007/s11042-020-09881-2.

[26] Jethanandani, M., Sharma, A., Perumal, T. and Chang, J.R., 2020. Multi-label classification based ensemble learning for human activity recognition in smart home. Internet of Things, 12, p.100324. doi: 10.1016/j.iot.2020.100324.

[27] Beimel, A., 1996. Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion. doi: 10.1109/ICASSP.1996.543514.

[28] Waters, B., 2011, March. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure

realization. In International workshop on public key cryptography (pp. 53-70). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-19379-2_6.

[29] Xia, Q., Gao, J., Obiri, I.A., Asamoah, K.O. and Worae, D.A., 2024. Selective Secure ABE Schemes Based on Prime Order Group. Proceedings of the International Conference on Cryptography and Network Security. doi: 10.1109/ICCS.2024.00041.

[30] Hosseini, S. and Seilani, H., 2021. Anomaly process detection using negative selection algorithm and classification techniques. Evolving Systems, 12(3), pp.769-778. doi: 10.1007/s12530-021-09392-7.

[31] Borlea, I.D., Precup, R.E. and Borlea, A.B., 2022. Improvement of K-means cluster quality by post processing resulted clusters. Procedia Computer Science, 199, pp.63-70. doi: 10.1016/j.procs.2022.01.009.

[32] Abualigah, L., 2020. Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications. Neural Computing and Applications, 32(16), pp.12381-12401. doi: 10.1007/s00542-019-05009-3.

[33] Emambocus, B.A.S., Jasser, M.B., Mustapha, A. and Amphawan, A., 2021. Dragonfly algorithm and its hybrids: A survey on performance, objectives and applications. Sensors, 21(22), p.7542. doi: 10.3390/s21227542

[34] Mirjalili, S., 2016. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27, pp.1053-1073. doi: 10.1007/s00542-015-2649-1.

[35] Alshinwan, M., Abualigah, L., Shehab, M., Elaziz, M.A., Khasawneh, A.M., Alabool, H. and Hamad, H.A., 2021. Dragonfly algorithm: a comprehensive survey of its results, variants, and applications. Multimedia Tools and Applications, 80, pp.14979-15016. doi: 10.1007/s11042-021-11394-z.

[36] Babcock, S., Beverley, J., Cowell, L.G. and Smith, B., 2021. The infectious disease ontology in the age of COVID-19. Journal of biomedical semantics, 12, pp.1-20. doi: 10.1186/s13326-021-00119-3.

[37] Zhou, D., Zhou, B., Zheng, Z., Kostylev, E.V., Cheng, G., Jimenez-Ruiz, E., Soylu, A. and Kharlamov, E., 2022, May. Enhancing knowledge graph generation with ontology reshaping–Bosch case. In European Semantic Web Conference (pp. 299-302). Cham: Springer International Publishing. doi: 10.1007/978-3-030-97343-2_34.

[38] Ma, R., Li, Q., Zhang, B., Huang, H. and Yang, C., 2024. An ontology-driven method for urban building energy modeling. Sustainable Cities and Society, 106, p.105394. doi: 10.1016/j.scs.2024.105394.

[39] Haykin, S., 1998. Neural networks: a comprehensive foundation. Prentice Hall PTR. doi: 10.1007/978-1-4615-1064-4.

[40] Farlow, S.J., 1984. Self-organizing Methods in Modeling (Statistics: Textbooks and Monographs, vol. 54). Marcel Dekker Inc. doi: 10.1145/1510043.1510076.

[41] Bazargani, M., H. Alizadeh, S. and Masoumi, B., 2024. Group deep neural network approach in semantic recommendation system for movie recommendation in online networks. Electronic Commerce Research, pp.1-40. doi: 10.1007/s12525-024-00426-1.

[42] Li, D., Armaghani, D.J., Zhou, J., Lai, S.H. and Hasanipanah, M., 2020. A GMDH predictive model to predict rock material strength using three non-destructive tests. Journal of Nondestructive Evaluation, 39, pp.1-14. doi: 10.1007/s10921-020-00710-4.

[43] Li, D., Moghaddam, M.R., Monjezi, M., Jahed Armaghani, D. and Mehrdanesh, A., 2020. Development of a group method of data handling technique to forecast iron ore price. Applied Sciences, 10(7), p.2364. doi: 10.3390/app10072364.

[44] Koopialipoor, M., Nikouei, S.S., Marto, A., Fahimifar, A., Jahed Armaghani, D. and Mohamad, E.T., 2019. Predicting tunnel boring machine performance through a new model based on the group method of data handling. Bulletin of Engineering Geology and the Environment, 78, pp.3799-3813. doi: 10.1007/s10064-019-01503-7.

[45] Armaghani, D.J., Momeni, E. and Asteris, P.G., 2020. Application of group method of data handling technique in assessing deformation of rock mass. 1, 1(1), p.001. doi: 10.1007/s12665-020-08892-x.

[46] Wazirali, R., 2020. An improved intrusion detection system based on KNN hyperparameter tuning and cross-validation. *Arabian Journal for Science and Engineering*, 45(12), pp.10859-10873. doi: 10.1007/s13369-020-04590-5.

[47] Ahakonye, L.A.C., Nwakanma, C.I., Lee, J.M. and Kim, D.S., 2021. Efficient classification of enciphered SCADA network traffic in smart factory using decision tree algorithm. *IEEE Access*, 9, pp.154892-154901. doi: 10.1109/ACCESS.2021.3122693.

[48] Deng, Z., Zhu, X., Cheng, D., Zong, M. and Zhang, S., 2016. Efficient kNN classification algorithm for big data. *Neurocomputing*, 195, pp.143-148. doi: 10.1016/j.neucom.2016.02.086.

[49] Izmailov, R., Vapnik, V. and Vashist, A., 2013, August. Multidimensional splines with infinite number of knots as SVM kernels. In *The 2013 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE. doi: 10.1109/IJCNN.2013.6706989.

[50] Kotsiantis, S.B., Zaharakis, I. and Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1), pp.3-24. doi: 10.1016/j.procs.2007.02.010.

[51] Mustaqeem, M. and Saqib, M., 2021. Principal component based support vector machine (PC-SVM): a hybrid technique for software defect detection. *Cluster Computing*, 24(3), pp.2581-2595. doi: 10.1007/s10586-021-03244-1.

[52] Tuttle, J.F., Blackburn, L.D. and Powell, K.M., 2020. On-line classification of coal combustion quality using nonlinear SVM for improved neural network NOx emission rate prediction. *Computers & Chemical Engineering*, 141, p.106990. doi: 10.1016/j.compchemeng.2020.106990.

[53] Zang, C. and Ma, Y., 2012. Ensemble machine learning: Methods and application. *Springer publication*, DOI, 10, pp.978-1-4471-2900-5. doi: 10.1007/978-1-4471-2901-2.

[54] Sun, X. and Qourbani, A., 2023. Combining ensemble classification and integrated filter-evolutionary search for breast cancer diagnosis. *Journal of Cancer Research and Clinical Oncology*, 149(12), pp.10753-10769. doi: 10.1007/s00432-023-04656-2.

[55] Tripathi, K., Khan, F.A., Khanday, A.M.U.D. and Nisa, K.U., 2023. The classification of medical and botanical data through majority voting using artificial neural network. *International Journal of Information Technology*, 15(6), pp.3271-3283. doi: 10.1007/s41870-023-01009-7.

[56] Liang, X., Cao, Z., Lin, H. and Shao, J., 2009, March. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 4th international symposium on information, computer, and communications security* (pp. 276-286). doi: 10.1145/1510043.1510076.

[57] https://www.kaggle.com/code/offmann/smart-home-dataset/data.

[58] https://www.kaggle.com/datasets/ranakrc/smart-building-system.

[59] https://www.kaggle.com/datasets/claytonmiller/open-smart-home-iotieqenergy-data?resource=download&select=Measurements.

[60] https://www.kaggle.com/datasets/claytonmiller/cubems-smart-building-energy-and-iaq-data.

[61] https://www.kaggle.com/datasets/claytonmiller/open-smart-home-iotieqenergy-data

[62] Melamed, I.D., Green, R. and Turian, J., 2003. Precision and recall of machine translation. In *Companion volume of the proceedings of HLT-NAACL 2003-short papers* (pp. 61-63). Association for Computational Linguistics. doi: 10.3115/1075096.1075110.