Chapter 12
Privacy-Preserving Analytics, Processing and Data Management

Kalmer Keerup, Dan Bogdanov, Baldur Kubo, and Per Gunnar Auran

Abstract Typically, data cannot be shared among competing organizations due to confidentiality or regulatory restrictions. We present several technological alternatives to solve the problem: secure multi-party computation (MPC), trusted execution environments (TEE) and multi-key fully homomorphic encryption (MKFHE). We compare these privacy-enhancing technologies from deployment and performance point of view and explain how we selected technology and machine learning methods. We introduce a demonstrator built in the DataBio project for securely combining private and public data for planning of fisheries. The secure machine learning of best catch locations is a web solution utilizing Intel® Software Guard Extensions (Intel® SGX)-based TEE and built with the Sharemind HI (Hardware Isolation) development tools. Knowing where to go fishing is a competitive advantage that a fishery is not interested to share with competitors. Therefore, joint intelligence from public and private sector data while protecting secrets of each contributing organization is an important enabler. Finally, we discuss the wider business impact of secure machine learning in situations where data confidentiality is a concern.

K. Keerup · D. Bogdanov · B. Kubo (✉)
Cybernetica AS, Narva mnt 20, 51009 Tartu, Estonia
e-mail: baldur.kubo@cyber.ee

K. Keerup
e-mail: kalmer.keerup@cyber.ee

D. Bogdanov
e-mail: dan.bogdanov@cyber.ee

P. G. Auran
SINTEF Digital, Strandvejen 4, Trondheim, Norway
e-mail: per.gunnar.auran@sintef.no

C. Södergård et al. (eds.), *Big Data in Bioeconomy*,
https://doi.org/10.1007/978-3-030-71069-9_12

12.1 Privacy-Preserving Analytics, Processing and Data Management

Data analysis and machine learning methods can provide great value in different areas of governance and business. By recognizing patterns in data, visualizing the patterns and developing predictive models, we can optimize farming, forestry and fishing operations.

Well-known data analysis and machine learning tools and frameworks can be used when the data originates from public sources such as Copernicus satellite images or from private sources when an agricultural business collects their own data. When data is confidential, current computers and software can protect data only while it is not being used or when data is being transferred. Typically, encryption and access restrictions are used. Traditional computers and software need to remove the technical protection to analyze data. Thus, the only protection of the owner of confidential data when using traditional software is limiting access to data to select few trusted persons and using contractual obligations.

One of the reasons for combining data from different companies and public sources is to improve the accuracy of machine learning and data analysis methods as data from different entities might capture different patterns or provide increased statistical power due to larger sample size. Learning from combined data can thus provide increased value for an industry. However, companies might be reluctant to share their data to protect the confidentiality of their operations.

Recently, secure computation technologies have been developed which enable processing confidential data without leaking individual values. By using these technologies, we are able to develop data analysis and machine learning software that retains the confidentiality of individual data providers but allows them to collectively gain improved insights from sharing their data.

When using secure computation, data is encrypted by the data owner and only then sent to a service processing the data. The host of the service will not have access to the unencrypted data nor the encryption keys. Data protection is not removed even while the data is being processed.

Secure computation technology can be used to develop solutions which are otherwise not possible due to confidentiality restrictions. There are some general types of problems where secure computation technology may be required:

- Outsourcing computations. Secure computation is a solution if one wishes to provide an analysis service to clients without learning the clients' data.
- Analyzing data governed by data protection laws. Secure statistical analysis can be used for decision-making when databases are governed by data protection laws and remain inaccessible for standard statistics software.
- Analyzing data from multiple sources. If data originates from a single provider, the provider can run analysis using their own infrastructure without giving data access to a third party. If we wish to analyze data from multiple sources without revealing the data to the party running the analysis, we can use secure computation technology.

In this chapter, we will describe two technologies for privacy-preserving data analysis and a demonstrator developed in the DataBio project which uses such technology to predict catch location and expected catch size for fisheries. The business impact of privacy-preserving data analysis and its applicability are also discussed.

12.2 Technology

Secure computation approaches can be categorized into software-based cryptographic techniques and hardware-based techniques. We bring examples from both categories.

12.2.1 Secure Multi-Party Computation

Secure multi-party computation (MPC) is a cryptographic technique for processing private data while preserving privacy. Sharemind MPC is a technology leveraging MPC which provides a framework for programming secure client–server applications. The roles of different parties involved in a Sharemind MPC process are as follows:

- Input parties who convert their public data into secret data and import it to servers hosted by computation parties.
- Computation parties who perform operations on the secret data without learning the input values or the results.
- Output parties who can retrieve the secret results from computation parties and construct the public result values.

Sharemind MPC uses an approach for MPC called additive secret sharing where private values are split into random values before being imported into an MPC system. This means that given a private 32-bit value x, two random values $x_{1,}x_2$ are generated and x_3 is computed so that $x \equiv x_1 + x_2 + x_3 \left(\mathrm{mod} 2^{32}\right)$. The three values are sent to three independent servers.

The servers can perform arithmetic on secret-shared values. For example, to add two values, each server adds their respective shares of the values. After the local additions, each server holds one share of the sum. More complicated operations require network communication between the servers. Figure 12.1 illustrates how two private values can be added using MPC.

As long as at most one of the servers is compromised, privacy remains protected. All three server hosts verify the analysis program before installing it. This ensures that only agreed upon results will be published to output parties. Shared responsibility also means that privacy remains protected if one of the servers is compromised. Sharemind MPC includes an auditing tool to detect tampering.

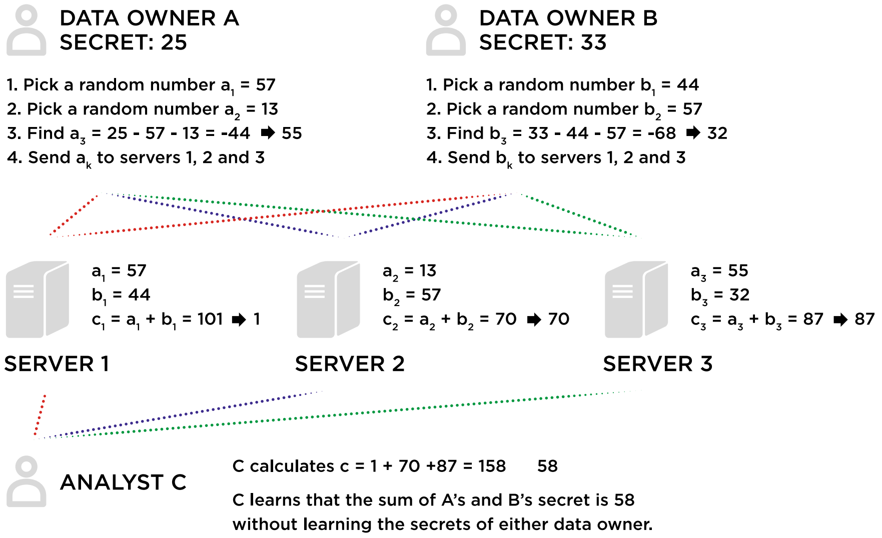

DATA OWNER A
SECRET: 25
1. Pick a random number $a_1 = 57$
2. Pick a random number $a_2 = 13$
3. Find $a_3 = 25 - 57 - 13 = -44$ → 55
4. Send a_k to servers 1, 2 and 3
DATA OWNER B
SECRET: 33
1. Pick a random number $b_1 = 44$
2. Pick a random number $b_2 = 57$
3. Find $b_3 = 33 - 44 - 57 = -68$ → 32
4. Send b_k to servers 1, 2 and 3
$a_1 = 57$
$b_1 = 44$
$c_1 = a_1 + b_1 = 101$ → 1
SERVER 1
$a_2 = 13$
$b_2 = 57$
$c_2 = a_2 + b_2 = 70$ → 70
SERVER 2
$a_3 = 55$
$b_3 = 32$
$c_3 = a_3 + b_3 = 87$ → 87
SERVER 3
ANALYST C
C calculates c = 1 + 70 +87 = 158 58
C learns that the sum of A's and B's secret is 58
without learning the secrets of either data owner.

Fig. 12.1 Illustration of adding secret-shared values

MPC is a general-purpose programmable technique and has been successfully used to implement practical applications [1]. The Sharemind MPC technology has been used for tax fraud detection [2], statistical analysis of government databases for a social study [3] and a report on the state of the Estonian IT industry by combining data from companies in the IT sector [4].

The main benefit of MPC is the high security guarantees. A party hosting an MPC server cannot learn anything about the values sent to it. There are no side-channel attacks which sometimes plague cryptographic techniques. Sharemind protects data in transit, in memory, at rest and during computations.

The main downsides of MPC are its complicated deployment requirements and decreased performance when compared to conventional software. Since the three server hosts must be independent, the organizations using MPC must decide on three parties who will be managing the servers. This involves more contracts between parties participating in the process when compared to a single organization providing an analysis service, but data will be protected technically, not just by the contracts as with usual data analysis tools.

12.2.2 Trusted Execution Environments

An alternative to software-based techniques is using a trusted execution environment such as Intel Software Guard Extensions (SGX).[1] SGX is an extension of the instruction set of Intel processors which enables developing secure applications when even the host operating system is not trusted. SGX relies on three concepts to protect data: enclaves, attestation, and data sealing.

SGX is a set of CPU instructions for creating and operating with memory partitions called enclaves. When an application creates an enclave, it provides a protected memory area with confidentiality and integrity guarantees. These guarantees hold even if privileged malware is present in the system, meaning that the enclave is protected even from the operating system that is running the enclave. With enclaves, it is possible to significantly reduce the attack surface of an application.

Remote attestation is used to prove to an external party that the expected enclave was created on a remote machine. During remote attestation, the enclave generates a report that can be remotely verified with the help of the *Intel attestation service*.[2] Using remote attestation, an application can verify that a server is running trusted software before private information is uploaded.

Data sealing allows enclaves to store data outside of the enclave without compromising confidentiality and integrity of the data. The sealing is achieved by encrypting the data before it exits the enclave. The encryption key is derived in a way that only the specific enclave on that platform can later decrypt it.

Sharemind Hardware Isolation (HI) is a technology using Intel SGX which provides the ability to process confidential data. Sharemind HI is built as a client–server service similar to Sharemind MPC. The client is an application that calls operations on the server, encrypts data and performs remote attestation on the server. The Sharemind HI server does the bulk of the work and is responsible for the following: checking if a user has the right to access the system; checking if a user has the correct roles to perform an operation; managing the encrypted user data and the encryption keys of the data; managing task descriptions of how a data analysis process is carried out; storing a log of the operations performed in the server and scheduling the tasks to run.

Figure 12.2 illustrates the security model of Sharemind HI applications. The input data, shown in red, is encrypted at the client side and sent to the server. The input data encryption keys of the data are securely transferred to the SGX protected enclaves. Likewise, the output data, shown in green, is encrypted inside of the enclave and stored on the server. When requested, the enclave securely transfers the output data encryption keys to the authorized clients.

At any point during the deployment, a client can request a cryptographic proof of what analysis code is running in the server, shown in blue on the figure. This proof can be compared against a previously generated proof by an auditor who has validated the code to be secure.

[1] Intel® Software Guard Extensions | Intel® Software.

[2] https://software.intel.com/en-us/sgx/attestation-services.

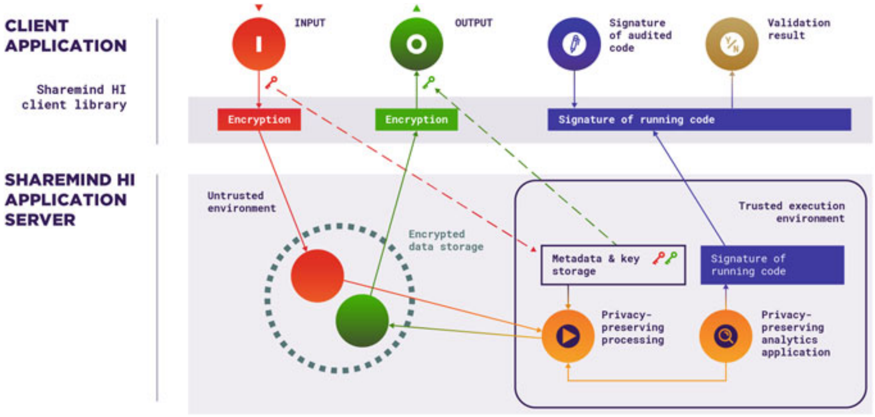

CLIENT APPLICATION
Sharemind HI client library
SHAREMIND HI APPLICATION SERVER
INPUT
OUTPUT
Signature of audited code
Validation result
Encryption
Encryption
Signature of running code
Untrusted environment
Encrypted data storage
Trusted execution environment
Metadata & key storage
Signature of running code
Privacy-preserving processing
Privacy-preserving analytics application

Fig. 12.2 Sharemind HI security model

The main benefits of Sharemind HI over Sharemind MPC are performance and simpler deployment. There is only one computational party, and unlike Sharemind MPC network communication is not required while the enclave is running.

Another benefit of Sharemind HI is that enclaves are programmed in the C++ programming language, whereas Sharemind MPC programs are written in a domain-specific language called SecreC which resembles C. This allows Sharemind HI programmers to adapt libraries and other existing code written in C or C++ .

The main downside of Sharemind HI is that it requires users to trust Intel. Details of how SGX-enabled processors are produced are undisclosed information, and Intel cannot prove that SGX is secure. It is also possible that side-channel attacks against SGX will be developed which would require more careful design of the enclave software. Practical applications should consider the security and performance trade-offs between cryptographic and hardware-based techniques.

12.2.3 Homomorphic Encryption

Another alternative for privacy-preserving computation is fully homomorphic encryption (FHE). FHE allows arbitrary computations on encrypted data. Privacy is ensured by encryption and is thus independent of the trustworthiness or security of the server that is executing the computation. See the UN Handbook on Privacy-Preserving Computation Techniques[3] for a summary of this family of encryption schemes.

[3] https://publications.officialstatistics.org/handbooks/privacy-preserving-techniques-handbook/UN%20Handbook%20for%20Privacy-Preserving%20Techniques.pdf.

12.2.4 On-The-Fly MPC by Multi-Key Homomorphic Encryption

One major disadvantage of classical MPC schemes (such as secret sharing) is that they need to be planned out in advance. The number of participants needs to be known and fixed before the calculation starts. In contrast, there is the concept of *on-the-fly MPC*, which is much more flexible in those regards. The main criteria an on-the-fly MPC scheme should meet are as follows:

1. The cloud can perform arbitrary, dynamically chosen computations.
2. It can use data from an arbitrary, non-pre-fixed set of participants (on-the-fly).
3. The computations are non-interactive, i.e., they do not require communication with all the participants (like with secret sharing).

On-the-fly MPC can be achieved by using multi-key fully homomorphic encryption (MKFHE). While most FHE schemes allow only one encryption key to be used, MKFHE schemes allow for multiple keys to be used for one computation.

Figure 12.3 illustrates how an MKFHE scheme can facilitate on-the-fly MPC. In this case, we have four different Alices with their secret message m_1, m_2, m_3 and m_4. Each of them encrypts their message using a different key (k_1, k_2, k_3 and k_4) and sends it to Bob. Out of these four encrypted messages, Bob can choose any subset (say Enc(m_1, k_1), Enc(m_2, k_2), Enc(m_3, k_3)) and any function that he wishes to perform on it (say f). Note that these choices can be made *after* the messages have been encrypted and sent to Bob.

He then calculates f(Enc(m_1, k_1), Enc(m_2, k_2), Enc(m_3, k_3)) and sends the result back to Alice1, Alice2 and Alice3, who agree to approve or disapprove the calculation. If approved, they can decrypt the result together and obtain $f(m_1, m_2, m_3)$. The decryption is only possible if the three of them work together. Note that there is no need for any communication with Alice4, since her message is not involved in the calculation. Also note that the other three Alices need not communicate until after Bob has finished his calculation. This gives MKFHE a huge advantage over classical MPC in terms of scalability and flexibility. However, like for other FHE schemes, the computation of f is very costly.

12.2.5 Comparison of Methods

All the methods discussed above have their advantages and disadvantages. The following table gives a rough overview.

Method	Advantages	Disadvantages
MPC by secret sharing	– Relatively efficient – Easy to handle – Already mature technology	– Requires coordinating multiple servers – Requires planning and setup

(continued)

(continued)

Method	Advantages	Disadvantages
Trusted execution environments	– High efficiency – Secure even if OS is not	– Vendor (Intel) proprietary technology that is not disclosed
Single-key homomorphic encryption	– Very flexible – Security independent of software and hardware – Needs only one server	– High computational cost – Difficult to understand/use – Allows for one key only
Multi-key homomorphic encryption	– Full flexibility – Security independent of software and hardware – On-the-fly execution	– High computational cost – Difficult to understand/use

For most practical use cases, computational cost (and thereby scalability) is by far the most important factor. The better flexibility that homomorphic encryption schemes offer may be crucial for some applications, but is generally less relevant. It was therefore decided that MPC and trusted execution environments would be feasible for the project.

12.3 Secure Machine Learning of Best Catch Locations

In order to demonstrate how secure computation technologies could be used in agriculture, forestry and fisheries, a demonstrator which predicts the best fish catch location and expected catch size on a given day was developed in the frame of the DataBio project.

Catch data with geographical positions was retrieved from the Norwegian Directorate of Fisheries [5]. Although we used public data for experimentation, our approach demonstrates that secure machine learning models can be trained on data from multiple fisheries and enables combining private data with public data.

12.4 Pipeline

In the pilot, we implemented the model using both Sharemind MPC and Sharemind HI [6]. Due to better performance, we chose the Sharemind HI solution as the backend for a web-based tool. The Sharemind MPC version is efficient enough to train models that can be reused for estimation afterward even if the model is kept private. As there are fishery-specific parameters, a model would need to be trained for each fishery. The Sharemind HI version trains a model in the order of a minute instead of hours it takes with Sharemind MPC.

Figure 12.4 illustrates the prediction pipeline using secure machine learning.

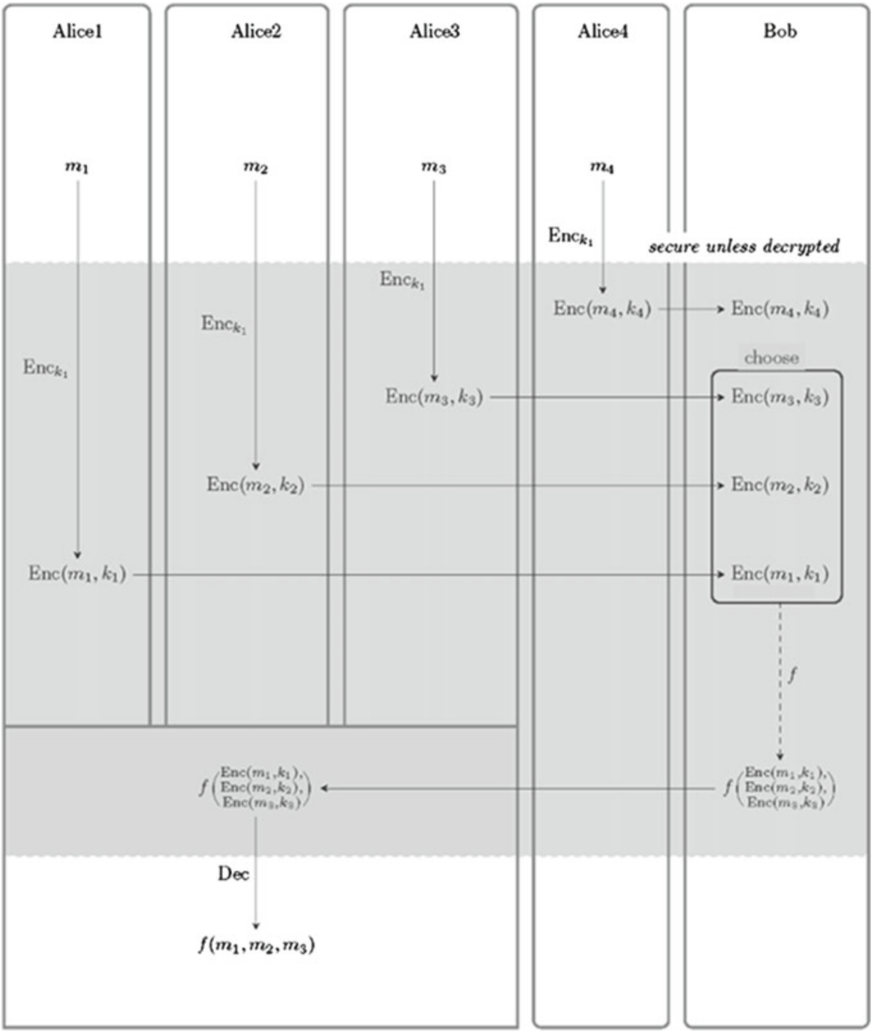


Fig. 12.3 On-the-fly MPC using an MKFHE scheme

The analysis takes into account the following parameters: harbor location, distance threshold, quantile of best catch, size of the ship and whether to maximize a single species catch or all species (total biomass output).

12.5 Model Development

Public catch data was used in the R[4] statistical analysis software to find a method for modeling the data. Since catch size and position vary by season, we could not use linear regression or autoregression for accurate prediction. A local regression method called LOESS was chosen due to its ability to model phenomena without a known function.

The program predicts three variables on a given date: latitude, longitude and catch size by fitting three LOESS regression models. LOESS is a nonlinear regression method which was developed for smoothing data. It allows one to see trends in scatterplots of noisy data.

LOESS trains a weighted linear regression model for each day by fitting a second-degree polynomial for local regression. The point estimated by the trained local model is given as the estimate for that day.

The user can specify a quantile argument to find the "best" catches to train LOESS models. For example, if the quantile argument is 0.9, then the top 10% data points by catch size are used for training the models. This means estimating where the best captains are fishing.

The user can also specify their home harbor and a distance threshold to filter out distant locations before fitting the model.

After choosing LOESS, we implemented fitting of LOESS models in both Sharemind MPC and Sharemind HI. We consider experimentation on public or generated data a good practice for finding a suitable model before implementing it using a secure computation technology.

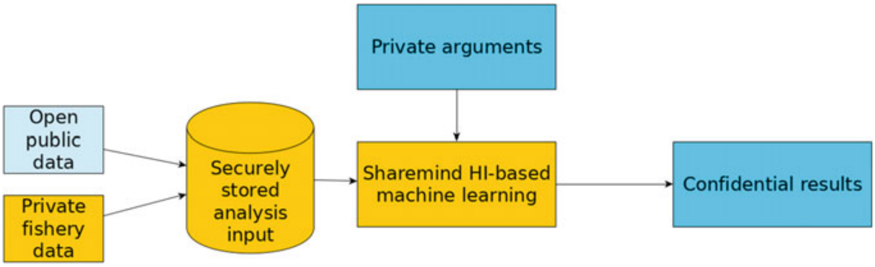
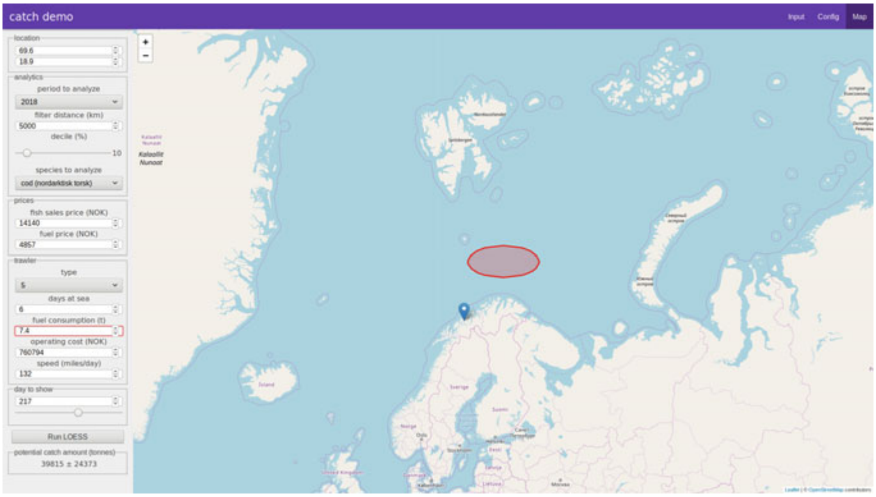


Fig. 12.4 Abstract overview of the proposed Sharemind HI-based solution

[4] R: T he R Project for Statistical Computing.



location
69.6
18.9

analytics
period to analyze: 2018
filter distance (km): 5000
decile (%): 10
species to analyze: cod (nordarktisk torsk)

prices
fish sales price (NOK): 14140
fuel price (NOK): 4857

trawler
type: 5
days at sea: 6
fuel consumption (t): 7.4
operating cost (NOK): 760794
speed (miles/day): 132

day to show: 217

potential catch amount (tonnes)
39815 ± 24373

Fig. 12.5 Catch location prediction demonstrator user interface

12.6 User Interface

A web-based interface was developed for the tool. It allows input parties to encrypt and import their data. Fisheries can use the tool to train the predictive model using their parameters.

The user can select the fish species, home harbor, distance threshold, vessel type and top catch quantile. After training the models, the enclave returns three vectors to the client application: latitude curve, longitude curve and catch size curve. The interface will display a map with the estimated position on a given day. The user can change the day with a slider to see how the position changes. The enclave also calculates prediction intervals for the fitted curves which allows the catch area to be displayed as an ellipse on Fig. 12.5.

12.7 Conclusions and Business Impact

The ability to handle confidential data in privacy-preserving analytics opens up for a number of new applications opportunities, not only in the fishery domain, but also in agriculture and forestry.

There are many situations where sensitive data is not made available because of concerns that the data becomes accessible by competitors or by others that might misuse the data.

The purpose of this demonstrator is to show that it is possible to handle confidential data as part of data analytics, potentially combining open data and confidential data in analytics that both provide business value and preserve data confidentiality. Confidential data with much higher precision on catch locations and time can be analyzed the same way, without the fishery shipping companies revealing to each other where they got the catches, resulting in a tool for catch prediction that all parties can benefit from to reduce time and energy costs looking for fish.

A wide business impact is foreseen by this demonstrator that shows that this is possible and a pipeline that can be reused in future applications where data confidentiality is a concern.

References

1. Archer, D. W., Bogdanov, D., Pinkas, B., & Pullonen, P. (2015). *Maturity and performance of programmable secure computation.* Cryptology ePrint Archive, Report 2015/1039, 2015. https://eprint.iacr.org/2015/1039
2. Bogdanov, D., Jõemets, M., Siim, S., & Vaht, M. (2015, January). How the Estonian tax and customs board evaluated a tax fraud detection system based on secure multi-party computation. In *International conference on financial cryptography and data security* (pp. 227–234). Springer.
3. Bogdanov, D., Kamm, L., Kubo, B., Rebane, R., Sokk, V., & Talviste, R. (2016). Students and taxes: A privacy-preserving study using secure computation. *Proceedings on Privacy Enhancing Technologies, 2016*(3), 117–135.
4. Talviste, R. (2011). Deploying secure multiparty computation for joint data analysis—a case study. *Master's thesis. University of Tartu. 2011.* https://sharemind.cyber.ee/files/papers/itl_app_talviste_2011.pdf
5. Fiskeridirektoratet. "Åpne data: fangstdata koblet med fartøydata." *Fiskeridirektoratet*, 2 Oct. 2018. https://www.fiskeridir.no/Tall-og-analyse/Aapne-data/Aapne-datasett/Fangstdata-koblet-med-fartoeydata
6. DataBio Hub Privacy-aware analytics. https://www.databiohub.eu/registry/#service-view/Privacy-aware%20analytics/0.0.1