Identifying Importation and Asymptomatic Spreaders of Multi-drug Resistant Organisms in Hospital Settings

Jiaming Cui[1], Jack Heavey[2], Eili Klein[3,4,5], Gregory R. Madden[6], Anil Vullikanti[2,7], and B. Aditya Prakash[1,*]

[1]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, US
[2]Department of Computer Science, University of Virginia, Charlottesville, VA 22904, US
[3]Center for Disease Dynamics, Economics & Policy, Washington, DC 20015, US
[4]Department of Emergency Medicine, Johns Hopkins School of Medicine, Baltimore, MD 21205, US
[5]Department of Epidemiology, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD 21205, US
[6]Division of Infectious Diseases & International Health, Department of Medicine, University of Virginia School of Medicine, Charlottesville, VA 22903, US
[7]Biocomplexity Institute, University of Virginia, Charlottesville, VA 22904, US

Supplementary Information

*To whom correspondence should be addressed. E-mail: badityap@cc.gatech.edu

Dataset

We extract three different types of patient data based on the electronic health records (EHR) from the University of Virginia hospital: patient demographic information and risk factors (e.g., comorbidities, medical history), lab testing, and contact network data.

Patient risk factor data

This dataset consists of risk factors for patients in intensive care units (ICUs). We extracted 19 different risk factors from the EHR for each patient, which are all available prior to MRSA testing. All patients received an MRSA test within $(t-3, t+3)$ days of being admitted into one of the ICUs, and patients who tested positive for MRSA within this range are considered as importation cases. From July 1, 2019, to December 31, 2019, 2470 patients were in the UVA ICUs, with 157 identified as importation cases. The following provides a definition for each risk factor and the method of their derivation:

- **MRSA nares**: Binary variable indicating whether the ICU admission to be administered is a nares PCR test or a clinical culture test. Negative clinical culture tests should not be used to indicate the absence of MRSA.
- **Device (e.g., central line)**: Binary variable that indicates whether a patient had a catheter, tube, or blood line (defined as "device") administered to them in the UVA hospital system.
- **Age**: Integer value indicating the age of the patient in years.
- **Admit from ED**: Binary variable indicating that the type of admission to the hospital was defined as an emergency.
- **Admit from healthcare facility**: A variable in $-1, 0, 1$ indicating where the patient was admitted from. If the patient was admitted from a skilled nursing facility or a long-term acute care hospital, it is 1; if admitted from somewhere besides these facilities, it is -1. It is 0 if the admission location is unknown.
- **Prior discharge to facility**: A variable in $-1, 0, 1$ based on a history of a patient's prior discharges from the UVA hospital system. It is 1 if the patient had previously been discharged to a skilled nursing facility or a long-term acute care hospital and -1 if they had been previously discharged but not to one of these facilities. It is 0 if there is no prior discharge data for the patient.
- **Female sex**: Binary variable indicating that the gender of the patient is female.
- **Recent surgery**: Binary variable indicating whether the patient has had a surgery performed at any time in the UVA hospital system before the date of the ICU admission.
- **Male sex**: Binary variable indicating that the gender of the patient is male.
- **Device usage within 90 days**: Binary variable indicating whether a device has been administered to the patient within the 90 days prior to the ICU admission.
- **Surgery history within 90 days**: Binary variable indicating whether the patient has had a surgery performed in the UVA hospital system within the last 90 days of the ICU admission.
- **Surgery history 90 days ago**: Binary variable indicating if the patient has had surgery in the UVA hospital system more than 90 days prior to the ICU admission.
- **Device usage 90 days ago**: Binary variable indicating whether a device has been administered to the patient more than 90 days prior to the ICU admission.
- **Length of stay**: Integer value defining how long a patient's current stay in the hospital has been before the ICU admission.

- **MRSA contacts within 7 days**: Integer value defining the number of contacts with an individual whose known MRSA status was positive over the 7 days prior to the test administration, as defined by the daily contact networks.
- **MRSA contacts within 14 days**: Integer value defining the number of contacts with an individual whose known MRSA status was positive over the 14 days prior to the test administration, as defined by the daily contact networks.
- **Elective admission**: Binary variable indicating that the type of admission to the hospital was defined as elective.
- **Urgent admission**: Binary variable indicating that the type of admission to the hospital was defined as urgent.
- **Routine admission**: Binary variable indicating that the type of admission to the hospital was defined as routine.

Lab testing data

The dataset consists of infection information for each patient. To diagnose MRSA, two types of tests were used in the University of Virginia hospital: culture tests and polymerase chain reaction (PCR) tests. However, a negative culture test does not disqualify an individual from being infected with MRSA, so we concentrated on positive culture tests and both negative and positive PCR tests. For a specific patient p on a specific day t, $y_{p,t} = 1$ indicates that the patient tested positive on day t or their most recent previous test was positive. Conversely, $y_{p,t} = 0$ signifies that the patient tested negative on day t or their most recent test was negative.

Contact network data

In this work, the agent-based model (ABM) used in NeurABM (SIS-ABM model) also took the contact network data as input. This dataset consists of a series of contact networks comprising three different types of nodes: patients, healthcare workers (HCWs), and locations, and each network is for one day. On a given day t, if i and j are colocated at any time on day t in location l, an edge will exist between patient/HCW nodes i, j, and location node l. The colocation data is aggregated from the EHR. Specifically, we focus on unweighted and undirected graphs. Therefore, each graph is represented as an $N \times N$ symmetric binary adjacency matrix $\boldsymbol{A}_t$ s.t. $\boldsymbol{A}_t(i,j) = 1$ if nodes i and j are connected with an edge at time t, $\boldsymbol{A}_t(i,j) = 0$ otherwise. The node-set is shared across all graphs. Because of the nature of this data, individuals such as support staff or patient guests are not tracked, and thus are not included in the network. Additionally, HCW-HCW colocations are not tracked in rooms where care is not administered, such as break rooms.

SIS-ABM model

In this work, we use SIS-ABM [6] as the agent-based model in our NeurABM framework to simulate the spread of MRSA. SIS-ABM is "pathogen load-based", where each patient can be either *Susceptible* (S)

Supplementary Table 1: List of SIS-ABM model parameters

Notation	Description
α	Pathogen shedding rate
β	Disease infectivity
δ	Recovery probability
γ_p	Natural pathogen reduction rate on patient nodes
γ_H	Natural pathogen reduction rate on HCW nodes
γ_L	Natural pathogen reduction rate on location nodes
τ_{ijt}	Transfer ratio from node j to node i at time t

or *Carriage* (C), and the model keeps track of the pathogen load on all nodes. Such pathogen load-based models are widely used in HAI modeling [10, 4, 9, 3, 11, 7, 1]. We show the pseudocode for the model in Supplementary Procedure 1. The parameters are listed in the Supplementary Table 1. In SIS-ABM model, each node in the contact networks carries some amount of pathogens that changes over time. The exchange of pathogens among nodes is driven by the edges, which indicates the close contacts between these nodes. It further uses τ_{ijt} to represent the ratio of the pathogen being transferred from node j to node i at time t (or remaining if $i = j$). Specifically, based on the kinds of nodes of i and j, we have 8 kinds of transfer ratios: τ_{PP}, τ_{PH}, τ_{PL}, τ_{HP}, τ_{HH}, τ_{HL}, τ_{LP}, τ_{LH}. It also uses γ_p, γ_H, γ_L to denote the natural pathogen reduction rate on patient, HCW, and location nodes. Using the transfer ratios τ_{ijt}, reduction rates γ_i, and adjacency matrix $\boldsymbol{A}_t$, we can construct the transfer matrix $\boldsymbol{R}_t$, and write the pathogen load updates using a linear operation as in Supplementary Procedure 1 step 5. Here, $\boldsymbol{x}_t$ is the infection state vector at time t, α is the pathogen shedding rate for infected patients. Note that the column-sums of $\boldsymbol{R}_t$ are restricted to be less than or equal to 1 (i.e., $|\boldsymbol{R}_t\boldsymbol{l}_t|_1 \leq |\boldsymbol{l}_t|_1$), since the total amount of pathogen cannot increase after transfer. In SIS-ABM model, the probability of a patient in the carriage state increases with the amount of pathogen, which is formulated as a dose-response function $f(\cdot)$ in Supplementary Procedure 1 step 8. Once infected, the patient sheds α additional pathogen per timestep to his own load, which can later be transferred to neighbors (both people and locations) via edges in contact networks; this shedding continues until the patient recovers. Note that susceptible patients may still be colonized with the pathogen loads and spread them to others. The model also assumes that HCWs and locations that are non-infectable, while they can still act as pathways of pathogen transfer.

Supplementary Procedure 1 SIS-ABM model

Inputs: $\Theta = \{\alpha, \beta, \delta, \{\boldsymbol{A}_t\}_{t=1}^{T}, \{\tau_{ijt}\}_{i,j,t}\}$
Initialize infection-states $\boldsymbol{x}_1$ and loads $\boldsymbol{l}_1$
Compute $\boldsymbol{R}_t(i,j) = \begin{cases} \tau_{ijt}\boldsymbol{A}_t(i,j) & \text{if } i \neq j \\ \tau_{ijt}\gamma_i & \text{if } i = j \end{cases}$ for all t.
for $t = 1, \ldots, T$ **do**
    Update loads $\boldsymbol{l}_{t+1} = \boldsymbol{R}_t\boldsymbol{l}_t + \alpha\boldsymbol{x}_t$.
    for each patient i **do**
        if i is susceptible at time t (i.e., $\boldsymbol{x}_t(i) = 0$) **then**
            i gets carriage (i.e. $\boldsymbol{x}_{t+1}(i) = 1$) with probability $f(\boldsymbol{l}_t(i)) = \min\{1, \beta\boldsymbol{l}_t(i)\}$.
        else
            i gets susceptible (i.e. $\boldsymbol{x}_{t+1}(i) = 0$) with probability δ.
        end if
    end for
end for

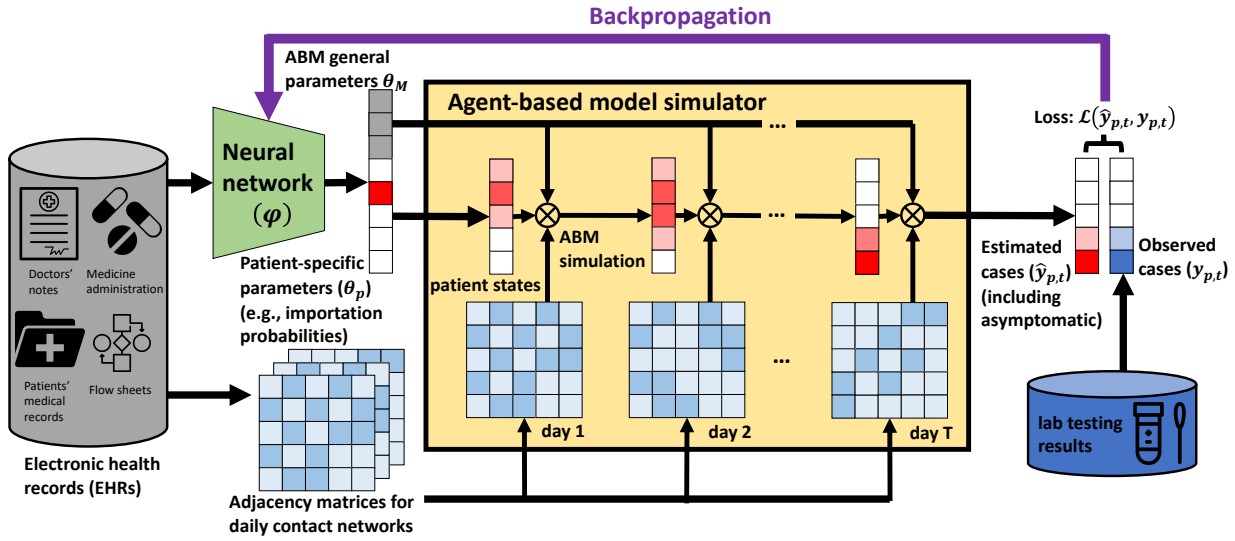
NeurABM framework

As shown in Supplementary Figure 1, the NeurABM is composed of two parts: the neural network part (green block, parameterized by ϕ) and the agent-based model simulator part (yellow block).

The neural network part takes the risk factors of patients $\boldsymbol{f}$ (where each element $f_p \in \boldsymbol{f}$ is for patient p) as input and then estimates both the patient-specific parameters $\boldsymbol{\theta}_p$ (where $\theta_p \in \boldsymbol{\theta}_p$ is for patient p, and in this work it is the probability that one patient is an importation case) and ABM parameters $\boldsymbol{\theta}_M$ (i.e., the general parameters that apply to every patient in the SIS-ABM model, including disease infectivity parameter β, recovery probability δ, pathogen shedding rate α, natural pathogen reduction rate for patients, HCWs, and locations γ_P, γ_H, γ_L, transfer ratios from different kinds of nodes τ_{PP}, τ_{PH}, τ_{PL}, τ_{HP}, τ_{HH}, τ_{HL}, τ_{LP}, τ_{LH}) together. Specifically, the neural network part can be represented by

$$\boldsymbol{\theta}_p, \boldsymbol{\theta}_M = NN(\boldsymbol{f}; \phi) \tag{1}$$

The ABM simulator part takes both the parameters from the neural network part (including both patient-specific parameters $\boldsymbol{\theta}_p$ and ABM parameters $\boldsymbol{\theta}_M$) and the adjacency matrices collected from EHR for each



Supplementary Figure 1: Our NeurABM framework involves 4 steps: (1) The neural network takes the patients' risk factor data collected from EHR as input, and outputs both the agent-based model (ABM) parameters, denoted by $\boldsymbol{\theta}_M$ (which are applicable to every patient), and patient-specific parameters, denoted by $\boldsymbol{\theta}_p$ (importation probabilities in this paper, darker red means higher probability), for each patient p. (2) These parameters and the adjacency matrices for contact networks of each day collected from EHR are then fed into the ABM simulator for simulation for T days, and the output will be the probability that each patient p is in the *Carriage* state $\hat{y}_{p,t}$ for each day t (darker red means higher probability). (3) We then compare this $\hat{y}_{p,t}$ with the ground-truth observation of patients in *Carriage* state (via lab testing), denoted by $y_{p,t}$, and compute the loss $\mathcal{L}(\hat{y}_{p,t}, y_{p,t})$. (4) We backpropagate this loss to the neural network to tune the neural network parameters ϕ.

day, $\boldsymbol{A}$, as input, and then runs the simulation process as shown in Supplementary Procedure 1. Specifically, since the probability that one patient is an importation case is included in $\boldsymbol{\theta}_p$, the patient states at day 0 are known. The simulation process can be repeated for arbitrary steps to simulate the MRSA spread in T days. The output is the estimated patient states on each day $\hat{\boldsymbol{y}}$. The ABM part can be represented by

$$\hat{\boldsymbol{y}} = ABM(\boldsymbol{A}; \boldsymbol{\theta}_p, \boldsymbol{\theta}_M) \tag{2}$$

With both the neural network part and the ABM simulator part, we can then pipeline both parts together to train them simultaneously. Specifically, one training epoch comprises the following four steps.

1. **Step 1:** The neural network takes risk factors of each patient as the input to estimate the patient-specific parameters $\boldsymbol{\theta}_p$ and ABM parameters $\boldsymbol{\theta}_M$ as in Supplementary Equation 1. Note that the importation probabilities are also included in the $\boldsymbol{\theta}_p$; thus, we can also get a vector of size N (N is the number of patients in the contact network) that represents the patients' states at $t = 0$.

2. **Step 2:** The ABM simulator then takes both the parameters from the neural network part (including patient-specific parameters $\boldsymbol{\theta}_p$ and ABM parameters $\boldsymbol{\theta}_M$) and the adjacency matrices $\boldsymbol{A}$ collected from EHR for each day as input to run simulations for T steps. The output will be the vector $\hat{\boldsymbol{y}}$ if size $N \times T$ as in Supplementary Equation 2, where $\hat{y}_{p,t}$ represents the probability of being in state *Carriage* for patient p on day t.

3. **Step 3:** We compare the estimated carriage probability $\hat{\boldsymbol{y}}$ with the corresponding ground-truth observations (i.e., observed carriage patients) $\boldsymbol{y}$. More specifically, we use the weighted binary cross-entropy loss (BCE loss)

$$\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sum_P \sum_T w_{pos} y_{p,t} \log(\hat{y}_{p,t}) + w_{neg}(1 - y_{p,t}) \log(1 - \hat{y}_{p,t}) \tag{3}$$

as the loss function. Here w_{pos} and w_{neg} are the weights for positive and negative observations. We set $w_{pos} : w_{neg} \propto \sum_P \sum_T \mathbb{1}[y_{p,t} = 0] : \sum_P \sum_T \mathbb{1}[y_{p,t} = 1]$, where $\mathbb{1}[\cdot]$ is the indicator function, which is 1 if the condition is true, and 0 otherwise.

4. **Step 4:** With the BCE loss $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$, noting that our ABM simulator part is also differentiable as described above, we can then calculate the gradient of the computed loss with respect to each parameter of the neural network via backpropagation. This allows us to better tune the neural network and learn more reasonable parameters as inputs for the ABM simulator.

We repeat the above 4 steps for several epochs until the loss $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ converges. The pseudo code is given in Supplementary Procedure 2.

Supplementary Procedure 2 NeurABM pseudo code

Inputs: Risk factors of patients $\boldsymbol{f}$, adjacency matrices of contact networks $\boldsymbol{A}$, lab testing results $\boldsymbol{y}$
for each epoch **do**
  Estimate the patient-specific parameters $\boldsymbol{\theta}_p$ and ABM parameters $\boldsymbol{\theta}_M$
  for each timestep t **do**
    Estimate the patient states on day t: $\hat{\boldsymbol{y}}_t$
  end for
  Compute the loss $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$
  Update the neural network parameters ϕ
end for
Outputs: Neural network parameters ϕ

The formal problem formulation is given as follows:

Problem Formulation

Given the risk factors of patients $\boldsymbol{f}$, adjacency matrices of contact networks $\boldsymbol{A}$, lab testing results $\boldsymbol{y}$, the neural network part as in Supplementary Equation 1, and the ABM part as in Supplementary Equation 2, find the best neural network parameter ϕ such that

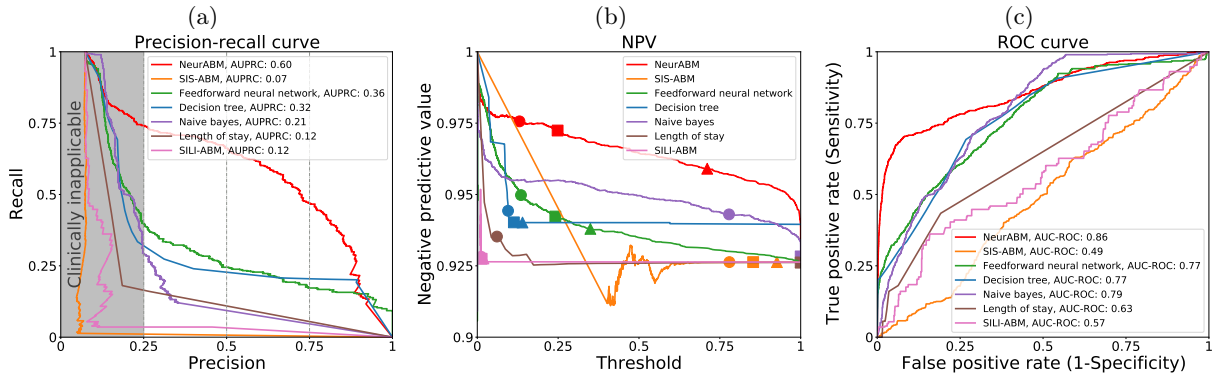
$$\arg\min_{\phi} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \mathcal{L}(ABM(\boldsymbol{A}; \boldsymbol{\theta}_p, \boldsymbol{\theta}_M), \boldsymbol{y}) = \mathcal{L}(ABM(\boldsymbol{A}; NN(\boldsymbol{f}; \phi)), \boldsymbol{y}) \tag{4}$$

With the trained NeurABM model, we can identify importation and nosocomial infection cases. For importation cases, note that the neural network part can estimate both patient-specific parameters $\boldsymbol{\theta}_p$ and ABM parameters $\boldsymbol{\theta}_M$. We can just use this $\boldsymbol{\theta}_p$ as the estimated probability of being an importation case for each patient. For nosocomial infection cases, we will use these ABM parameters $\boldsymbol{\theta}_M$ as well as the patient state vector on the last day of week $k-1$ to run simulations and estimate the patient states in week k.

Baselines

To compare our NeurABM with current modeling or machine learning-based methods, we also compare with other baselines including Feedforward neural network, decision trees, naive bayes, SIS-ABM, SILI-ABM, clinical heuristic methods, and so on. For machine learning-based methods, we train two models: one for identifying importation cases, and another for identifying nosocomial infection cases.

- SIS-ABM: As an ablation study, we use only the SIS-ABM model to identify importation and nosocomial infection cases to see how it performs.
- SILI-ABM: Sequential, individual-level inference (SILI) model [9] is an agent-based model that uses Bayesian approaches to infer the colonization probability of confirmed carriers for each patient.
- Feedforward neural network: Feedforward neural networks have shown good performance in different clinical estimation and prediction tasks in recent years. In this work, we also implemented a multi-layer perceptron (MLP) feedforward neural network using Scikit-learn [8] in Python.

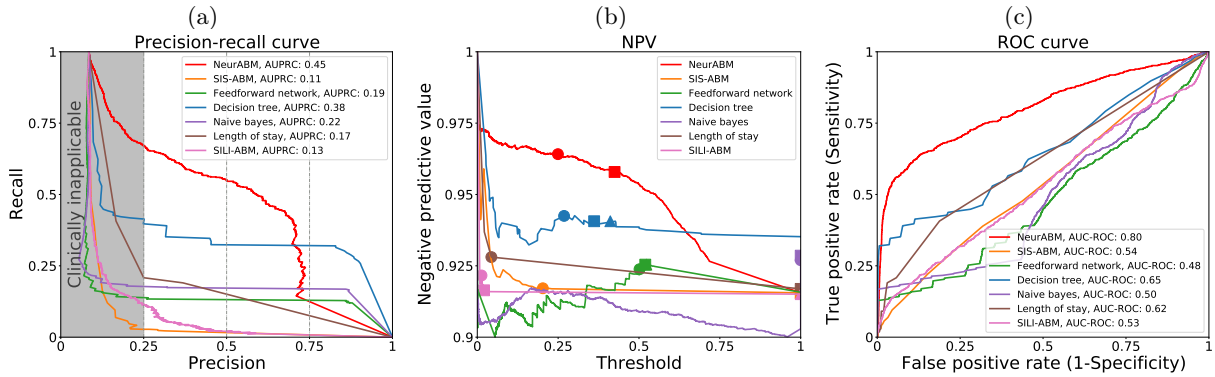


(d) Identifying MRSA importation cases

Method	Precision	Recall (Sensitivity, True positive rate)	F1 score	AUPRC	False positive rate (1-Specificity)	Negative predictive value (NPV)	AUC-ROC
NeurABM	0.25	0.74	0.37	**0.60**	0.18	0.98	**0.86**
	0.50	0.66	0.57		0.05	0.97	
	0.75	0.47	0.58		0.01	0.96	
SIS-ABM	0.25	0.01	0.02	0.07	0.01	0.93	0.49
	0.25	0.01	0.01		0.01	0.93	
	0.75	0.01	0.01		0.01	0.93	
Feedforward neural network	0.25	0.40	0.31	0.36	0.10	0.95	0.77
	0.50	0.24	0.33		0.02	0.94	
	0.75	0.17	0.28		0.01	0.94	
Decision tree	0.25	0.32	0.28	0.32	0.08	0.94	0.77
	0.50	0.23	0.31		0.02	0.94	
	0.75	0.20	0.32		0.01	0.94	
Naive bayes	0.25	0.29	0.27	0.21	0.07	0.94	0.79
	0.50	0.09	0.16		0.02	0.93	
	0.75	0.05	0.09		0.01	0.93	
Length of stay	0.25	0.16	0.20	0.12	0.04	0.94	0.63
	0.50	0.11	0.18		0.03	0.93	
	0.75	0.05	0.10		0.02	0.93	
SILI-ABM	0.25	0.04	0.06	0.12	0.02	0.93	0.57
	0.50	0.03	0.06		0.01	0.93	
	0.75	0.02	0.03		0.01	0.93	

Supplementary Figure 2: The performance in identifying importation cases includes: (a) The precision-recall curves (PRC). The x-axis represents precision, and the y-axis represents recall. The red and other color curves represent NeurABM and other baselines. A larger area under the precision-recall curve (AUPRC) indicates better performance. AUPRC values are listed in the legends, and NeurABM has the highest AUPRC value. (b) The negative predictive value (NPV) with different thresholds. The x-axis is the threshold for classification, and the y-axis is the NPV value. Circles, squares, and triangles correspond to the thresholds and NPV values where precision is 0.25, 0.5, and 0.75, respectively. A higher NPV value indicates fewer missing importation cases that are not identified and therefore better performance, and NeurABM has the highest NPV values. (c) The receiver operating characteristic (ROC) curves in identifying MRSA importation cases. The x-axis is the false positive rate, and the y-axis is the true positive rate. A larger area under the ROC (AUC-ROC) indicates better performance. AUC-ROC values are listed in the legends, and NeurABM has the highest AUC-ROC value. (d) The recall, F1 score, AUPRC, false positive rate, NPV, and AUC-ROC under different precisions (0.25, 0.5, 0.75). The best AUPRC and AUC-ROC are in bold.

- Decision tree: Decision tree is also a widely-used machine learning algorithm in classification and prediction [2]. It represents choices in a tree-like graph, illustrating different outcomes as branches. In this work, we also use a decision tree as a baseline using Scikit-learn [8] in Python.
- Naive bayes: Naive bayes is also a powerful and efficient algorithm in machine learning [5]. It operates on the principle of Bayes theorem, where "naive" assumes that features are independent. Similarly, we use Scikit-learn [8] and implemented a naive bayes classifier.
- Length of stay: As suggested by a previous study [9], patients with longer hospital stay days are suspected of having a higher risk of infection. Therefore, we use the EHR to calculate the days a



(d) Identifying nosocomial infection cases

Method	Precision	Recall (Sensitivity, True positive rate)	F1 score	AUPRC	False positive rate (1-Specificity)	Negative predictive value (NPV)	AUC-ROC
NeurABM	0.25	0.67	0.36	**0.45**	0.19	0.96	**0.80**
	0.50	0.55	0.52		0.05	0.96	
	0.75	0.12	0.21		0.01	0.92	
SIS-ABM	0.25	0.03	0.05	0.11	0.01	0.92	0.54
	0.50	0.02	0.03		0.01	0.92	
	0.75	0.01	0.02		0.01	0.92	
Feedforward neural network	0.25	0.14	0.18	0.19	0.04	0.92	0.48
	0.50	0.13	0.21		0.02	0.93	
	0.75	0.13	0.22		0.01	0.93	
Decision tree	0.25	0.41	0.31	0.38	0.11	0.94	0.65
	0.50	0.32	0.39		0.03	0.94	
	0.75	0.32	0.45		0.01	0.94	
Naive bayes	0.25	0.18	0.21	0.22	0.05	0.93	0.50
	0.50	0.17	0.26		0.02	0.93	
	0.75	0.17	0.28		0.01	0.93	
Length of stay	0.25	0.21	0.23	0.17	0.06	0.93	0.62
	0.50	0.15	0.23		0.03	0.92	
	0.75	0.08	0.14		0.01	0.92	
SILI-ABM	0.25	0.11	0.16	0.13	0.03	0.92	0.53
	0.50	0.02	0.04		0.01	0.92	
	0.75	0.01	0.02		0.01	0.92	

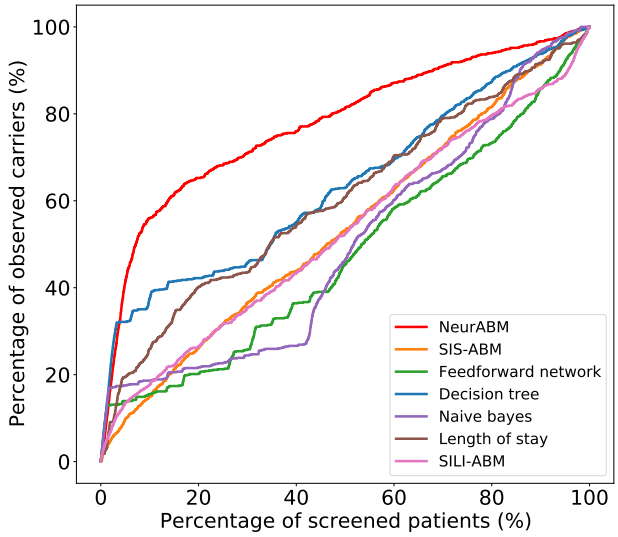
Supplementary Figure 3: The performance in identifying nosocomial infection cases includes: (a) The precision-recall curves (PRC). The x-axis represents precision, and the y-axis represents recall. The red and other color curves represent our NeurABM and other baselines. A larger area under the precision-recall curve (AUPRC) indicates better performance. AUPRC values are listed in the legends, and NeurABM has the highest AUPRC value. (b) The negative predictive value (NPV) with different thresholds. The x-axis is the threshold for classification, and the y-axis is the NPV value. Circles, squares, and triangles correspond to the thresholds and NPV values where precision is 0.25, 0.5, and 0.75. A higher NPV value indicates fewer missing nosocomial infection cases that are not identified and therefore better performance, and NeurABM has the highest NPV values. (c) The receiver operating characteristic (ROC) curves in identifying MRSA nosocomial infection cases. The x-axis is the false positive rate, and the y-axis is the true positive rate. A larger area under the ROC (AUC-ROC) indicates better performance. AUC-ROC values are listed in the legends, and NeurABM has the highest AUC-ROC value. (d) The recall, F1 score, AUPRC, false positive rate, NPV, and AUC-ROC under different precisions (0.25, 0.5, 0.75). The best AUPRC and AUC-ROC are in bold.

patient stays in the hospital, and use it as the clinical heuristic baseline for comparison.

Additional experiment results

In addition to the baselines listed in the main article, we also compared with SILI model [9] in identifying importation cases. As shown in Supplementary Figure 2, our NeurABM (red) achieves better performance than the SILI model (pink).

We also ran the experiments where we trained until week $k-2$ and identified nosocomial infection cases



Supplementary Figure 4: Percentage of identified MRSA cases by screening "high-risk" patients. For each patient in the UVA ICUs, we use each method to estimate their MRSA infection probability and rank them according to this probability from high to low. We then screen different percentages of patients (x-axis) and see how many actual MRSA cases can be identified (y-axis). As seen in the figure, NeurABM can always identify more MRSA cases than other baselines.

in week k, and the results show that NeurABM still performs better than other baselines.

As shown in Supplementary Figure 3a, the x-axis and y-axis represent precision and recall, respectively. The red curve represents our NeurABM framework. As shown in the figure, the area under the precision-recall curve for our framework is the largest (0.45) compared to other baselines. The dashed grey lines correspond to the precision of 0.25, 0.5, and 0.75. Again, our NeurABM always achieves the highest recall with precision equal to 0.25, 0.5, and 0.75, indicating that our framework is effective. In Supplementary Figure 3b, we show how the negative predictive value (NPV) changes with threshold changes. We can see that the NPV rate is always higher than 0.95 and other baselines, indicating that our NeurABM can identify nosocomial infection cases well with fewer missing/undetected patients. We also show the ROC curve in Supplementary Figure 3c. Here, the area under the precision-recall curve for our framework is the largest (0.80) compared to other baselines. In the table in Supplementary Figure 3d, NeurABM always achieves the highest recall and F1 score with a given precision, demonstrating the effectiveness of our framework in identifying nosocomial MRSA infection cases.

To better examine the performance of our NeurABM method and other baselines in identifying MRSA cases for control, we show that if we test the patients based on the ranked infected probability estimated by each method, what percentage of MRSA cases can be identified among all cases. As shown in Supplementary Figure 4, we rank all patients according to the estimated infected probability of each method from high to low. We then test different percentages of patients and see how many actual MRSA patients can be identified by each method. Here, we can see that NeurABM can always identify more nosocomial MRSA infection cases (y-axis) given the same test budget (x-axis), which suggests that our framework is effective and practical in identifying MRSA cases in clinical settings.

References

[1] Adhikari, B., Lewis, B., Vullikanti, A., Jimenez, J. M., and Prakash, B. A. Fast and near-optimal monitoring for healthcare acquired infection outbreaks. *PLoS computational biology 15*, 9 (2019), e1007284.

[2] Bishop, C. Pattern recognition and machine learning. *Springer google schola 2* (2006), 5–43.

[3] Brouwer, A. F., Weir, M. H., Eisenberg, M. C., Meza, R., and Eisenberg, J. N. Dose-response relationships for environmentally mediated infectious disease transmission models. *PLoS computational biology 13*, 4 (2017), e1005481.

[4] Chandrasekaran, S., and Jiang, S. C. A dose response model for staphylococcus aureus. *Scientific Reports 11*, 1 (2021), 1–10.

[5] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[6] Jang, H., Justice, S., Polgreen, P. M., Segre, A. M., Sewell, D. K., and Pemmaraju, S. V. Evaluating architectural changes to alter pathogen dynamics in a dialysis unit. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2019), IEEE, pp. 961–968.

[7] Nelson, R. E., Jones, M., Leecaster, M., Samore, M. H., Ray, W., Huttner, A., Huttner, B., Khader, K., Stevens, V. W., Gerding, D., et al. An economic analysis of strategies to control clostridium difficile transmission and infection using an agent-based simulation model. *PloS one 11*, 3 (2016), e0152248.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[9] Pei, S., Liljeros, F., and Shaman, J. Identifying asymptomatic spreaders of antimicrobial-resistant pathogens in hospital settings. *Proceedings of the National Academy of Sciences 118*, 37 (2021), e2111190118.

[10] Plipat, N., Spicknall, I. H., Koopman, J. S., and Eisenberg, J. N. The dynamics of methicillin-resistant staphylococcus aureus exposure in a hospital model and the potential for environmental intervention. *BMC infectious diseases 13*, 1 (2013), 1–11.

[11] Rubin, M. A., Jones, M., Leecaster, M., Khader, K., Ray, W., Huttner, A., Huttner, B., Toth, D., Sablay, T., Borotkanics, R. J., et al. A simulation-based assessment of strategies to control clostridium difficile transmission and infection. *PloS one 8*, 11 (2013), e80671.